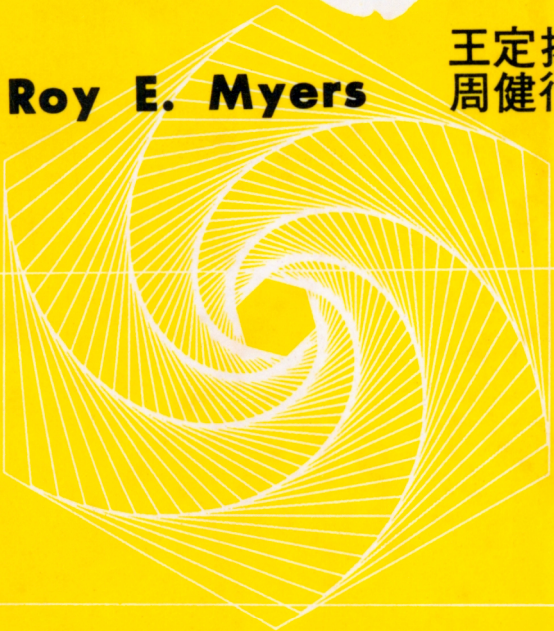
APPLE II 電腦製圖

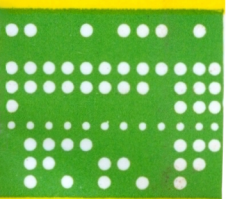
Microcomputer GRAPHICS

Roy E. Myers

王定揚 周健行 編譯



復文書局



APPLE II
電腦製圖

Microcomputer GRAPHICS

Roy E. Myers 王定揚 周健行 編譯

復文書局

APPLE Ⅱ電腦製圖

著作權執照台內著字第　　號

版權所有　　翻印必究

中華民國七十二年七月初版發行

特價 134 元

編譯者：王定揚・周健行

發行者：吳主和

發行所：復文書局

地址：臺南市東門路421巷28號

門市部地址：台南市林森路二段63號

電話:(062)370003・386937

郵政劃撥帳戶32104號

No.28. LANE421 DONG-MEN ROAD TAINAN TAIWAN REPUBLIC OF CHINA

TEL:(062)370003・386937

本書局經行政院新聞局核准登記發給
出版事業登記證局版台業字第0370號

譯者序

Apple微電腦的旋風正如火如荼的吹遍了全台灣，我們眼見小學生熟練的操作及專注的神情，中學生置身致力於TV GAME的樂趣，人們更配合著APPLE Ⅱ的功能從事研究軟體應用的發展，而造成今天APPLE Ⅱ 在全國上下男女老幼所掀起的一股熱潮與震憾。我們倖能恭逢其時，而在APPLE Ⅱ多功能的世界裏，特對彩色繪圖顯示的能力，投其所好進有拍案叫絕之樂。試想：當微電腦繪出五彩繽紛的生動畫面之際，能不教人心動，而歎其奧妙，讚其絢麗於置身其間？

然而坊間書店，實難找一本適合於繪圖設計或程式的好書。原文書尚少，中文書幾無，灰心之刻。所幸皇天不負苦心人，有見Roy E. Myers著的“Microcomputer Graphics”一書，興起萬思感激之情。爲期奇文共賞，我們投下心力，遍尋資料，費下不少時日，本著“口語化”之前提來編譯此書，不僅完成我們之所願，更盼與我們同好的國人共同深入其繪圖領域的探討。畢竟“寸步有芳鄰”是我們自始至終的確信。

在本書的編譯上，我們儘量地完全配合作者的原意。書裏的原理說明、程式分析與技巧均是今日“電腦繪圖”的重要考據。基於此我們不惜費時試驗了每一程式，並擷取珍貴的執行結果，以輔助原書之缺少，而最大的盼望是讀者的喜愛及興趣。全書採由淺而深、由基礎入設計的目標向讀者介紹，有點線圓的組合，亦有矩陣、向量應用數學的探討及平面、立體圖形的解剖以至完成封面封底六個複雜圖形的高段技巧。因此我們愼重選擇此書以作我們對APPLE Ⅱ 深入研究系列的第一本書籍，是見我們的重視與祈盼。

這在衆APPLE 書籍裏是一種突破的推出，只希望我們是APPLE Ⅱ「微電腦繪圖」的耕耘者，讀者是收穫及喜悅者。最後謹在此萬分感謝國立高雄工專電子科主任廖斌毅先生所給予多方面的協助與校對，更感謝本班同窗與親友們給予我們的指導與鼓勵，值得一提的是吾友張志誠、郭懷忠、陳永發、忻鼎竑等位同學的熱心幫助加以摯友鄭恪、蔡雅惠女友的備加詢問，致使我們能有此微末成品呈獻予讀者們，謹致最誠摯的謝意。譯者才疏學淺，又匆促付梓，遺漏疏忽之處，尚祈諸位先進學者不吝賜正爲盼！

譯者　王定揚
　　周健行　謹識於高雄工專

前　言

亞里斯多德（ Aristotle ）將圖畫在沙灘上。而達文西（ Leonardo Da Vinci ）用墨水畫圖來解說其在工程和解剖結構方面的研究。在能善用言語表達之前，小孩子就可以利用鉛筆或蠟筆將其感情和想法表達在紙上。不論是使用那一種媒體，長久以來，圖像的價值即已經被肯定。即所謂“千言萬語不如一張圖畫”。

目前，電腦已經廣泛應用在控制汽車，立體音響，及太空船等方面。此時，若說圖像的發展與顯示不使用電腦的話，那倒眞教人覺得驚訝。其實，電腦繪圖的應用早已經從電視遊樂器發展到電腦輔助設計，以及顯示由航海家二號（ Voyager 2 ）通過土星時送回地球的圖像。

電腦繪圖的應用，始自 1950 年代。自此以後，發展非常快速，不過熟悉電腦繪圖者卻爲數不多。在幾年前，由於繪圖設備的昂貴價格令一般電腦使用者敬而遠之。而幾年後的今天，一部完整的微電腦繪圖系統只要幾萬元台幣即可購得。不僅工商教育機構，甚至連個人也負擔得起，並深感其妙用無窮，而享受其充滿趣味性及挑戰性之樂。

廉價的微電腦繪圖系統雖無法與大型電腦相抗衡，但對一般水準的程式設計者而言，它已經能在合理的時間內提供合理的圖像。由於微電腦的繪圖能力比不上比較複雜的繪圖系統，因此微電腦的程式設計者就必須設法解決其先天上在繪圖硬體與繪圖語言方面的限制。

本書主要目的是要提供在微電腦繪圖中所必須的數學理論與程式設計的技巧。且這些數學理論適用於任何電腦系統。至於程式雖是針

對Apple Ⅱ使用的BASIC 語言而寫，不過有很多部分還是可以適用於其他的微電腦系統。

本書並不打算教授讀者如何設計BASIC 程式。我們假設讀者已經熟悉這種語言，並曾閱讀過“Applesoft BASIC程式參考手册”及“Apple Ⅱ參考手册”二書。另外書中的程式例乃是假設讀者的Apple Ⅱ爲48 K 的系統，並最好附有磁碟機，不過大部分的程式還是可以在較小系統上執行。若沒有磁碟機的話，也可以使用卡帶來代替。由於記憶容量較小的Apple Ⅱ 系統將只提供一個高解像圖形頁。因此須更改HGR2 指令爲HGR:POKE－16302,0 方能使程式正常動作。

同時，本書的程式例題是用來說明繪圖技巧而非設計程式的技巧；因此，這些程式並非是完美的範例，不過聰明如您的設計者，必能設法改善其執行時間而使程式更有效率，更完美。書中所有程式皆已試驗及偵錯過，若執行結果與書中所言不符，則必是您輸入程式時，按鍵錯誤所致。

致謝

獨力工作的成果畢竟有限，唯有與他人交往接觸方能獲益最多。有感於此，我深知好友Herb Mckinstry 之不惜奉獻其時間和智慧之精神實令我受益非凡。其與另一好友Tom Dwyer的建議，對我非常有幫助。同時也感謝Neal ，Alice ，Karen ，Glenn 和Linda 等人啓發我撰寫此書的動機和靈感。更要感謝Karen 在準備材料上的協助，以及Karen和Alice 的幫忙校稿。

程式磁碟

本書所有的程式皆已錄製在一片磁碟上，另外還加上一些有關立體圖和消除隱藏線等須耗時計算的機器語言副程式。這片磁碟大約是美金 15 元，若有興趣可聯繫：

Roy Myers
Graphics Programs Disk
R.D.1 , Box 3150
Leechburg , PA 15656

所複製的磁碟都是 16-sector 的，除非您特別註明要 13-sector 的磁碟。

原書作者簡介

Roy E. Myers 作者是美國賓夕凡尼亞州立大學的副教授。作者對微電腦軟體亦曾多次的揭露和發表且也有在兩家電腦化影片生產公司工作的經驗。

目　錄

第一部份　緒論……1
第一章　微電腦的繪圖世界……3
光域與光點……3
摘述……15
注意與建議……15
第二部份　電腦……17
第二章　APPLE Ⅱ的繪圖特性……19
記憶體的分配與使用……19
記憶體的定址顯示……22
彩度……31
解像度……32
直線的定義……34
HCOLOR 設定……35
軟性開關……37
記憶體搬運……41
其它繪圖頁……45
第 2 文字頁……47
黑白反倒圖像……49
程式設計助手……50
注意與建議……52
第三章　繪圖幕文字設計的技巧……55
點圖型設計的技巧……55

造型表設計的技巧…… 67
第三部份　平面圖形…… 77
第四章　電腦繪圖的軟性技巧…… 77
顯示幕比例 …… 77
畫面的截割及觀測台…… 78
變換…… 83
第五章　解析幾何與電腦繪圖……107
座標軸……108
直線……110
圓……113
橢圓……119
參數型方程式……126
拋物線……128
雙曲線……129
極座標曲線……132
函數型……134
第四部份　立體圖形……137
第六章　立體圖形的基本原理……139
座標系統……139
變換……143
立體物體的平面圖形……152
第七章　隱藏線和隱藏表面的處理……173
緒論……173
測定表面的方位……174
兩個物體；塗黑法……189
兩個物體；二分找尋法……191
兩個物體；任意的觀測點……193
第五部份　動態……221

第八章　動態圖形……223
用HPLOT 畫動態圖形……223
旋轉方塊……224
動態的立體圖形……226
造型表的動態圖形……231
第六部份　週邊設備和軟體……235
第九章　系統的構成要件……237
輸入裝置的硬體和軟體……237
輸出裝置的硬體和軟體……241
附　錄……245
附錄1　二進制，十進制與十六進制……247
二進制……248
十六進制……249
十六進制與二進制的關係……252
附錄2　造型表……261
圖形與造型表……261
程式A2.2 的研究……265
造型表的使用法……267
程式A2.2 的圖案（圖形組合）……270
附錄3　向量……275
向量的純量乘算……276
直線的向量表示……276
三度空間內的向量……278
向量的點積……279
向量的交積……280
半平面……281
附錄4　矩陣……285
矩陣的乘積……285

附錄 5　角的弧度測量……289
附錄 6　封面圖形的程式……291
封面的左上圖……291
封面的右上圖……296
封面的左下圖……297
封面的右下圖……300
封底的左圖……300
封底的右圖……302
圖 5.1……305
附錄 7　APPLE 指令一覽表 ……307
APPLESOFT 指令……307
DOS 指令 ……308
中英文名詞對照……309

程式例題索引

1.1 星際……4
1.2 跳球……5
1.3 太空船……6
1.4 波紋圖案……7
1.5 彩色線條……9
1.6 曲線……9
1.7 圓……10
1.8 螺旋條紋……11
1.9 美術家的助手……13

2.1 點圖型法……28
2.2 點圖型字元……30
2.3 彩度……31
2.4 混合彩度……32
2.5 更高解像度……33
2.6 HCOLOR 設定……35
2.7 HCOLOR 控制的改良……36
2.8 閃爍X及其方塊……39
2.9 動態方塊……40
2.10 記憶體搬運的副程式……43
2.11 姓名複印……44
2.12 其它繪圖頁……46

2.13 兩個文字頁…… 49

3.1 點圖字型…… 62
3.2 繪圖頁的點圖型文字…… 64
3.3 造型表字型檔…… 68
3.4 繪圖頁的造型表文字…… 71

4.1 截割…… 81
4.2 矩形之一…… 83
4.3 矩形之二…… 83
4.4 矩形變化…… 84
4.5 矩陣…… 87
4.6 均質座標…… 99
4.7 六邊形……103
4.8 巢串六邊形……105

5.1 直線……110
5.2 直線的截割……111
5.3 直線變量……112
5.4 圓之一……113
5.5 圓之二……115
5.6 圓之三……117
5.7 圓的截割……112
5.8 橢圓之一……121
5.9 橢圓之二……123
5.10 橢圓之三……125
5.11 參數型……127
5.12 拋物線……129

5.13 拋物線的旋轉……130
5.14 雙曲線……132
5.15 極點……133
5.16 函數軌跡……134

6.1 立方體……160
6.2 觀測點……162
6.3 變換立方體……167
6.4 表面……169

7.1 消除隱藏線（測定表面方位法）……186
7.2 消除隱藏線（兩個物體；塗黑法）……189
7.3 消除隱藏線（兩個物體；二分找尋法）……191
7.4 消除隱藏線（兩個物體；任意觀測點）……209
7.5 表面……214
7.6 隱藏線（對Z＝F（X，Y）形式的表面）……216

8.1 旋轉方塊……225
8.2 旋轉角錐……228
8.3 尺蠖……231

9.1 搖桿的讀取……237
9.2 電子筆……238
9.3 改良的電子筆……239

A1.1 十六進制轉換十進制……252
A1.2 十進制轉換十六進制……252

A2.1 雷達……269
A2.2 圖形的組合……270

A6.1 封面左上圖；APPLE Ⅱ……292
A6.2 封面右上圖；餘弦函數的表面……296
A6.3 封面左下圖；表面／平面……298
A6.4 封面右下圖；魔咒標誌……300
A6.5 封底左圖；圓形圖表……301
A6.6 封底右圖；房屋……302
A6.7 圖5.1；心臟圖……305

第一部份

緒　論

2

第一章　微電腦的繪圖世界

“電腦繪圖”對每個人來說具有不同的含意。這怎麼說呢？有人認爲它意味了電視遊樂器的雛形；也有其他的人作航空機訓練的模擬；建築師認爲它提供了建築物藍圖設計及顯示的方法；而工程師則利用了電腦繪圖來設計及分析他們的電子網路和電橋實驗。

因此人們利用電腦來繪圖的目的有很多且是因人而異。而此書的目的主要針對世面上廉價微電腦上的繪圖像所須注意的數學觀念及寫程式的技巧來加以分析。本書所採用的微電腦是目前世界（尤其是國內）上正如痴如狂風行的機種——APPLE Ⅱ，因此我們討論 Apple Ⅱ的基礎而且本書所有程式都是用 Applesoft BASIC 語言來設計。而本章，我們希望以程式例題的分析而對電腦與程式處理二者間的功能加以說明，至於程式敍述與技巧將在以後的數章分別討論。

光域與光點（Rasters And Pixels）

APPLE 是用光域（rasters）來顯示，簡單的說就是由水平線（光域）的聚集所形成的顯示幕，而每一條光域線所集合的點叫做光點（pixels）。舉例來說，APPLE 的高解像繪圖幕，是有 192 條光域線，而每一光域線包含了有 280個光點。光域線的編號是從 0 到 191；而光點是在每一光域線的 0 到 279。因此每個光點要有兩個數來決定。由光域線數目的數及每條光域線的光點的數。爲何向讀者介紹光域和光點的觀念呢，實在是繪圖幕上的圖像都是由許多光點的亮與暗所決定。

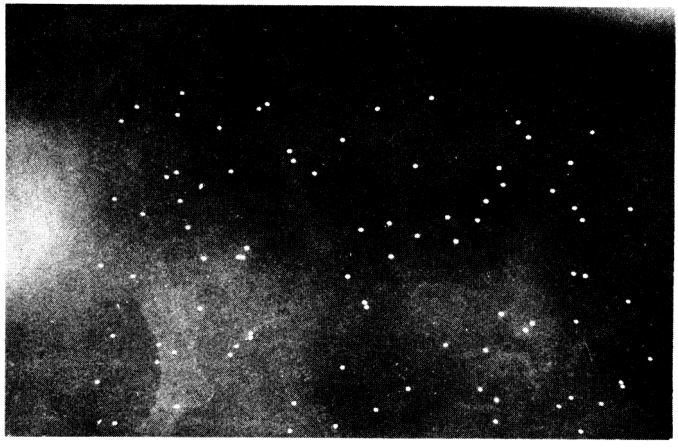
在光域掃描電腦繪圖的敍述中有必須能使個別的光點呈“亮”的

能力（即畫點）。Applesoft 裏HPLOT命令可執行此任務。HPLOT X,Y將光域線 Y中的X位置的光點呈“亮”。例如，HPLOT 0,0 使顯示幕左上角的光點發亮；HPLOT 279,191 使顯示幕右下角的光點發亮。因此使用此命令可輕易的控制任何光域任何位置的點。

從此以後大部份的圖案都是由很多個別點所組合而成。例如我們第一個例題程式就僅利用HPLOT命令所發展的圖案。

程式1·1：星際

```
1  REM PROGRAM 1.1 (SPACE)
2  REM PLOTS RANDOMLY SELECTED POINTS
10  HOME : HGR : HCOLOR= 3
20  FOR I = 1 TO 100
30 X = RND (1) * 279:Y = RND (1) * 191
40  HPLOT X,Y
50  NEXT I
```

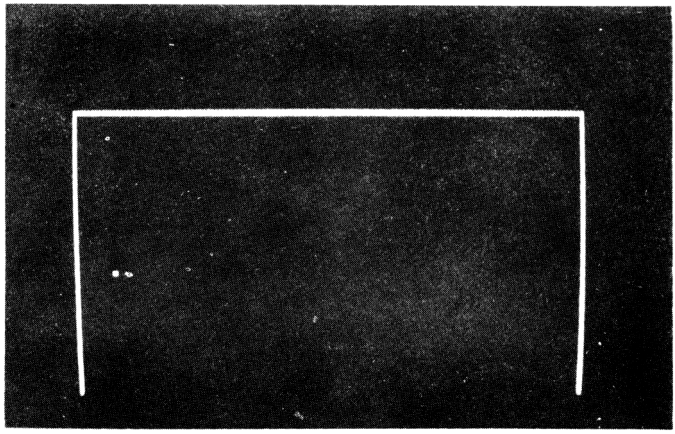


研究上面的程式，顯示星際的“星星”是由行號30 隨機選擇其位置的；而每當程式執行一次，就會顯示一個新的“星星情景”之顯示幕。

第二程式主要還是HPLOT 命令的練習，試試皮球跳動的模擬：

程式 1·2：跳球

```
1  REM PROGRAM 1.2 (BOUNCING BALL)
2  REM SIMULATES A BALL BOUNCING OFF FLOOR AND
   WALLS
10 DX = 4:X = 8:Y = 0:A = 2:V = 0
20  HOME : HGR
30  HCOLOR= 2: HPLOT 2,0 TO 2,133 TO 266,133
    TO 266,0
40  HCOLOR= 3: HPLOT X,Y: HPLOT X + 1,Y
50  IF Y = 132 THEN V = - V: IF V = 0 THEN V =
    - 20
60  IF X > 263 THEN DX = - DX
70  IF X < 6 THEN DX = - DX
80 V = V + A
90  HCOLOR= 0: HPLOT X,Y: HPLOT X + 1,Y
100 Y = Y + V:X = X + DX
110  GOTO 40
```



“球”的出現來自行號40 爲了使“球”看得更清楚，程式並肩地畫兩個點。行號 30 設計了使球反覆跳彈於“牆”與“地板”之間，至於球的運動，水平的運動是常數DX 來控制，垂直的運動是變數V及加速度A來控制。行號 40 則是白點球運動的模擬，而行號 90 則是擦去此球。行號 100是將點（X，Y）改變而畫上新位置。使球反

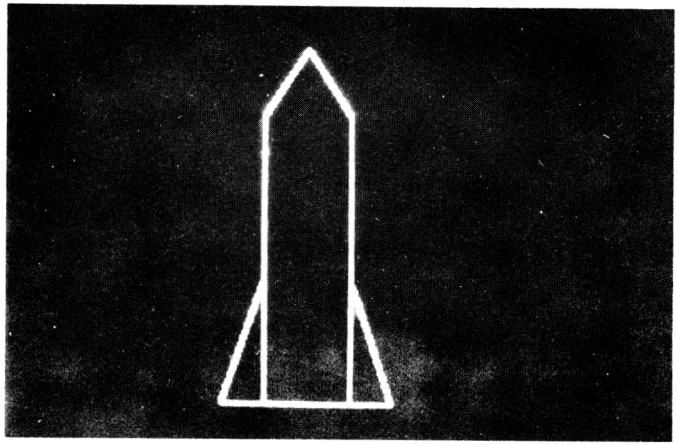
跳離開那兩道“牆”及“地板”的控制是從行號50到70。當球的反跳在很低情形（ Y = 132 和V = 0 ）時行號50 則將此球運動加速。若欲讓動作的中斷可用 CTRL C 或 RESET 。

> 注意：本章，程式所使用的技巧，本書後面章節會詳盡敍述，請各位讀者不要斟酌目前程式的細節，享受我們所提供的繪圖樂趣吧！

下面的程式是依照孩子們的說法所繪的太空船。

程式 1·3：太空船

```
1  REM PROGRAM 1.3 (SPACE SHIP)
2  REM USES HPLOT TO DRAW ELEMENTARY SHAPES
10  HOME : HGR2 : HCOLOR= 3
20  HPLOT 120,170 TO 120,50 TO 140,20 TO 160,50
    TO 160,170
30  HPLOT 160,120 TO 180,170 TO 100,170 TO
    120,120
```



由於經驗上的領會向讀者肯定此類程式對繪圖的顯示是有其參考價值的，但在此程式的寫作之前讀者得先將模型概略地先畫在繪圖紙頁上，方可得到所須之圖案。

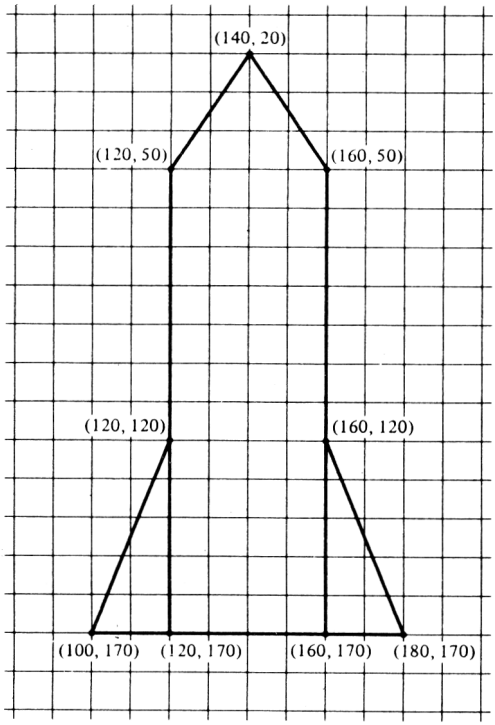


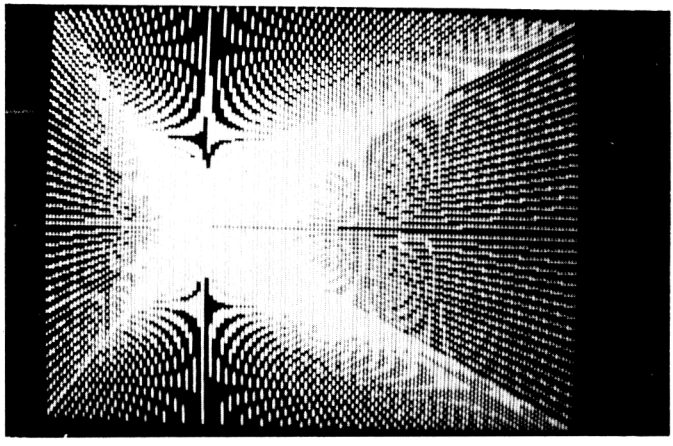
圖 1-1　太空船

現在將話題再轉向我們這些“設計家”程式。

程式 1·4 ：波紋圖案

```
1   REM PROGRAM 1.4 (MOIRE)
2   REM DRAWS OVERLAPPING RANDOM PATTERNS
10   HGR2 :A = 5
20 X1 = RND (1) * 279:Y1 = RND (1) * 191
30 C = C + 1: IF C = 8 THEN C = 0
40   HCOLOR= C:Y = 191
50   FOR X = 0 TO 279 STEP A
60   HPLOT X,191 TO X1,Y1 TO 279 - X,0
70   NEXT X
80   FOR Y = 0 TO 191 STEP A
90   HPLOT 279,191 - Y TO X1,Y1 TO 0,Y
100   NEXT Y
110   FOR I = 1 TO 500: NEXT I
120   IF PEEK ( - 16384) = 155 THEN POKE
      - 16368,0: TEXT : HOME : END
130   GOTO 20
```

欲停止程式執行，請按ESC鍵。



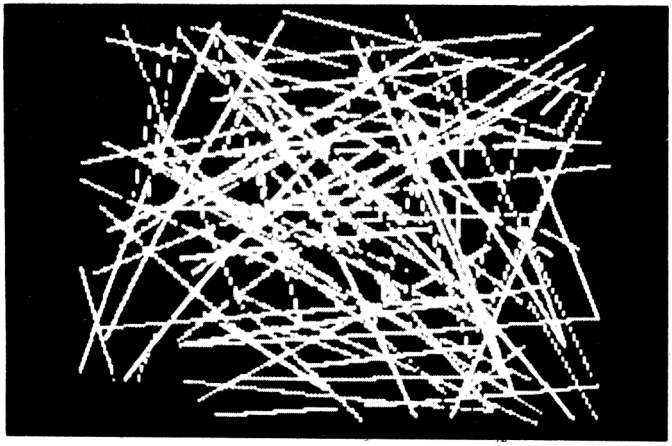
對於本MOIRE程式主要在建立Apples 彩色選擇的改變，其效果的產生以後的章節會詳加討論，目前，讀者只須儘量去玩味它！

行號20 是在隨機的情況下定圖案的“中心”。此程式畫連續的圖案於彼此間，行號30，40可改變色彩。行號110 設計了一個停頓的延遲方法，行號120則是離開程式的另一種新的方法；壓 ESC 鍵則程式結束，等到第2章時會討論到更多更妙的技巧，等待吧！

下面又是一個更簡單及更簡化的圖案。

程式 1·5 ：彩色線條

```
1  REM PROGRAM 1.5 (COLORED LINES)
2  REM RANDOMLY DROPS COLORED LINES ON THE
   SCREEN
10  HGR2
20 X1 = RND (1) * 279:Y1 = RND (1) * 191
30 X2 = RND (1) * 279:Y2 = RND (1) * 191
40 C = C + 1: IF C = 8 THEN C = 0
50  HCOLOR= C
60  HPLOT X1,Y1 TO X2,Y2
70  GOTO 20
```

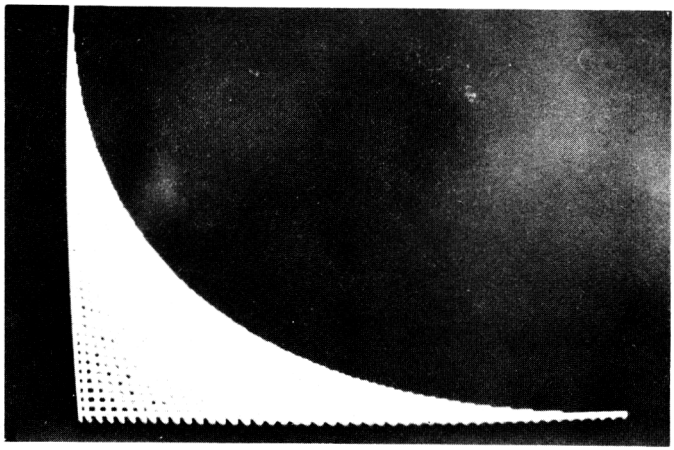


行號 20 及 30 選擇了要畫行號 60 時的線段端點。而這連續的線段將用不同的顏色來設計。

至今，我們的例題裏用到了點及直線，但在繪圖的領域裏，尚有劃曲線及類似直線的線段（微彎線）。下面就與讀者介紹此類的例題技巧：

程式 1·6 ：曲線

```
1  REM PROGRAM 1.6 (CURVE)
2  REM STRAIGHT LINES - - > CURVE
10  HGR2 : HCOLOR= 3
20  FOR Y = 0 TO 191 STEP 4
30  HPLOT 0,Y TO 279 * Y / 191,191
40  NEXT Y
```

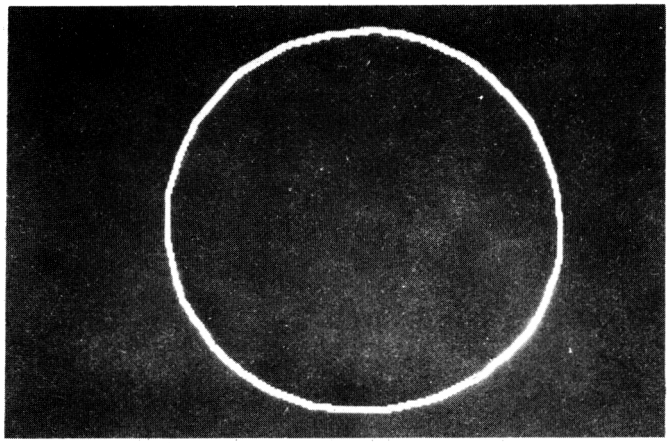


很明顯的，只有行號 30 畫直線段，而最後的圖像是一接近直線的曲線，而就好像“細線雕塑”的體裁。

圓的生成在第五章時會利用數種技巧來完成，在這裏我們使用那些技巧中速度最快的方法來討論。至於 SC，CX，CY分別的意義則在第四章來說明了。

程式 1·7：圓

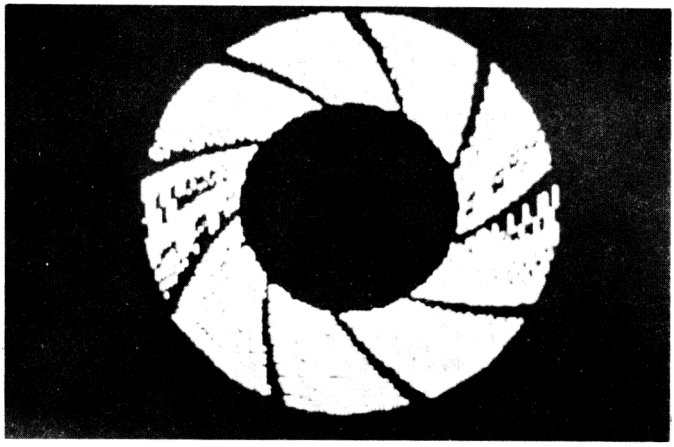
```
1  REM PROGRAM 1.7 (CIRCLE)
2  REM CIRCLE GENERATOR
10 DT = .1:C = COS (DT):S = SIN (DT)
20 SC = 1.16:CX = 140:CY = 96
30  HGR2 : HCOLOR= 3
40 X = 0:Y = 80
50  HPLOT SC * X + CX,Y + CY
60  FOR I = 1 TO 63
70 T = X * C - Y * S:Y = Y * C + X * S:X = T
80  HPLOT TO SC * X + CX,Y + CY
90  NEXT I
```



興趣的讀者若苦心研究圓的程式則會另設計出一生動螺線的圓條紋。同前所述，技巧的討論將留在以後討論。

程式 1·8：螺旋條紋

```
1   REM PROGRAM 1.8 (RING)
2   REM DRAWS A COLORFUL CIRCULAR BAND
10 DT = .1:C = COS (DT):S = SIN (DT)
20 SC = 1.16:CX = 140:CY = 96
30   HGR2 : HCOLOR= 3
40 X = 0:Y = 80
50   HPLOT SC * X + CX,Y + CY
60   FOR J = 1 TO 20
70   FOR I = 1 TO 63
80 T = X * C - Y * S:Y = Y * C + X * S:X = T
90   READ CL: POKE 28,CL: IF CL = 255 THEN
     RESTORE
100   HPLOT TO SC * X + CX,Y + CY
110   NEXT I
120 Y = Y - 2: NEXT J
130   DATA 42,85,127,128,170,213,255
```



此章最後的例題也是最複雜的例題。而這最後介紹的程式將是告訴讀者如何經由鍵盤與電腦的相互溝通來產生的圖案。

此程式的主角是一有顏色的移動“游標（cursor）”其游標將由讀者按[C]，[D]或[E]鍵來決定其將在的模式（modes）。在C模式時，僅見游標而已；在D模式時，則游標會畫滿線段於其所在每一位置；在E模式時，游標位置將由黑色線段所畫滿，即有效的擦去顯示幕的線段（因爲背景顏色是黑色）。

在D模式時顏色的考慮亦由1，2，3，4，5，或6的按鍵來決定，而各數目的顏色請看Applesoft參考手册。

游標的大小可按[L]鍵來增長；按[S]鍵來縮短。而要改變的速度加快可同時按 [REPT] 及 [L] 或 [S] 鍵即可。

最後游標四周位置點的改變可由讀者按[I]，[J]，[K]或[M]來選擇。它們變化結果如下：

[I]	上移
[J]	左移
[K]	右移

M　下移

同時壓住 I ， J ， K 或 M ，及 REPT 鍵時，可連續改變其位置。

程式所利用到的便利方法是造形表。在例題裏，行號10到行號40 定義了一簡單的造形表，游標也是利用程式來控制其大小、旋轉、顏色及位置。至於造形表的討論在Applesoft參考手册（第9章），而本書的附錄2亦有說明。

程式1·9：美術家的助手

```
1   REM PROGRAM 1.9 (ARTIST'S AID)
2   REM PROVIDES PAINTBRUSH UNDER KEYBOARD
    CONTROL
10   POKE 768,1: POKE 770,4
20   POKE 771,0: POKE 772,4
30   POKE 773,0
40   POKE 232,0: POKE 233,3
50   HGR : POKE - 16302,0:C = 3:S = 25:
     X = 140:Y = 96
60   FOR I = 1 TO 64
70 A = PEEK ( - 16384): POKE - 16368,0
80   IF A = 155 THEN TEXT : HOME : END
90   IF A = 177 THEN C = 1
100   IF A = 178 THEN C = 2
110   IF A = 179 THEN C = 3
120   IF A = 180 THEN C = 4
130   IF A = 181 THEN C = 5
140   IF A = 182 THEN C = 6
150   IF A = 202 THEN X = X - 1: IF X < 0
      THEN X = 279
160   IF A = 203 THEN X = X + 1: IF X > 279
      THEN X = 0
170   IF A = 201 THEN Y = Y - 1: IF Y < 0
      THEN Y = 191
180   IF A = 205 THEN Y = Y + 1: IF Y > 191
      THEN Y = 0
190   IF A = 211 THEN S = S - 1: IF S < 1
      THEN S = 1
200   IF A = 204 THEN S = S + 1
210   IF A = 195 THEN F = 0
220   IF A = 196 THEN F = 1
230   IF A = 197 THEN F = 2
240   IF A = 200 THEN HGR2
250   HCOLOR= C: ROT= I: SCALE= S
260   IF F = 0 THEN HCOLOR= 3: XDRAW 1 AT X,Y:
      XDRAW 1 AT X,Y: GOTO 300
```

```
270  DRAW 1 AT X,Y
280  IF F = 1 THEN 300
290  HCOLOR= 4: DRAW 1 AT X,Y
300  NEXT I
310  GOTO 60
```

請讀者小心打字。很多因打字上的錯誤使結果顯示

?SYNTAX ERROR

而當讀者完成了此程式的打字後，再打 R U N 及按 RETURN 鍵。則近顯示幕中心處的地方立刻呈現一閃爍的旋轉棒。

準備你的電腦畫筆來玩遊戲吧：

- 按D鍵進入D模式畫讀者欲畫之圖。
- 同時地按 K 和 REPT 鍵約二秒（顯示出寬的向右衝程（stroke））。
- 按 5 鍵改變顏色。
- 同時地按 M 和 REPT 鍵則顯示向下的拂掠（sweeping）衝程。
- 按 C 鍵則停止畫，游標可見但移動時不畫其軌跡。
- 同時地按 I 和 REPT 鍵則向左移動。
- 同時地按 S 和 REPT 鍵則游標縮短。
- 現在按 D 鍵，我們繼續開始畫圖了。

發揮讀者的想像和創造的能力吧！按 ESC 鍵可中止程式，而使程式回到文字頁（text page）及結束。

如果讀者想回到繪圖頁去歎賞你的作品，不要打HGR因爲它是回到高解像繪圖頁後又立即擦掉。應該打字：

```
POKE -16297,0: POKE -16302,0: POKE -16304,0:
POKE -16300,0
```

如果讀者想儲存圖樣以便以後的鑑賞，則可以儲存在磁碟裏。而首先決定此影像的NAME，然後打字：

```
BSAVE NAME, A$2000, L$2000
```

如此任何時刻想看儲存影像：打字

```
HGR: POKE -16302,0: PRINT CHR$(4); "BLOAD NAME,
A$2000"
```

摘　述（Summary）

本章所討論產生繪圖像的短程式例題。雖然是不非常老練，使用的方法或許不很明顯。但因由於其技巧會在以後章節中解釋，所以當你讀完本書後，你將會有長足的進步，而這些程式你會感到如此而已。

注意與建議

1. 本章有很多程式都使用了RND　命令。讀者也許會認爲它確實能產生“隨機（ random ）”數目（亂數系列）。其實，電腦開機每次一開始所提供的結果數目都是一樣，因此雖叫“隨機”圖案，事實上有好幾個程式所形成的圖案是可以預知的。爲使程式完全達到“隨機”數目。試著打入下面程式後再結合程式 1.1 ， 1.4 及 1.5

```
3 TEXT: HOME
4 PRINT "WHEN READY, PRESS ANY KEY TO START"
5 IF PEEK (-16384)>127 THEN POKE -16368,0:
  GOTO 10
6 X=RND (1)
7 GO TO 5
```

程式行號 5 — 7 會將電腦的隨機數目的結果不停地讀取數目而直到鍵被壓爲止。而由於按鍵的時間無法預期，因此，緊接 RND　命令所執行的數目亦無法預期。如此便達到了隨機亂數的功能。程式使用到的兩個命令 PEEK(−16384) 和 POKE(−16368,0　) 的功能在第 2 章會討論。

2. 繪圖顯示也可利用聲音使程式更生趣活潑。因爲 Apple 裏有一揚聲器，我們何不使用它呢？程式 1.2 的跳球實驗時將行號 50 修改如下，將會有聽得見的反彈球聲。

```
50 IF Y=132 THEN S=PEEK (-16336): PEEK (-16336):
   V=- V: IF V=0 THEN V= - 20
```

3. 加深熟悉繪圖的顯示不妨多畫一些類似程式1.3太空船的簡單圖樣。例如你們自己英文姓名的起首字母或全部姓名、風景（山，樹）、房屋、及城市中以天空爲背景而映出的建築物、山、樹等之輪廓等等。首先畫像在繪圖紙，然後轉換程式的點與線以畫圖案在繪圖顯示幕上。

4. 試程式1.4的改變。行號10定A值爲5，以控制產生圖案的線條的濃密度。改寫行號10定A＝3，4，6或7。

5. 如果讀者厭倦程式的打字，可利用磁碟來儲存，當然本書的所有程式都可使用。請翻回前磁碟儲存敍述的說明。我們也特別爲讀者準備了一塊含有這本書所提到的所有程式的磁碟片，盼您善加利用，欲購方式請閱前言。

第二部份

電　腦

18

第二章　APPLE Ⅱ的繪圖特性

記憶體的分配與使用 （Memory Allocation And Usage）

APPLE Ⅱ的繪圖輸出實際是顯示電腦記憶體區域的內容。基於此特性，我們得開始研究 Apple Ⅱ記憶體空間的分配與使用。標準的使用法要詳看圖 2.1 的記憶位置表（ memory map ），使用者可以利用這些記憶位置有效的使用電腦。而一般記憶位置爲讓讀者明瞭起見，常使用十六進制（前加“$”）和十進制一起表示。

> 注意：本章自始至終，都用簡單的例題以說明原理之下來討論。所以最好是在讀本書看程式時讀者手邊能有一部 Apple Ⅱ，如此更可事半功倍之效。

Apple Ⅱ的主板上有 48K RAM和 16K ROM 的記憶體，因此總共有 64K的記憶體。一般 K 相當於 1000 但在電腦領域裏 K 表示 1024 。因此 48K 意味 48(1024) = 49152 ，而 64K表示 64(1024) = 65536。這些數值正是記憶體可利用的數目，但由於電腦是一個二進制的機器，65536 相當二進制的 10000000000000000, 而我們又習慣於十進制甚過二進制，這點的差異，也是學電腦者須克服的。

記憶體中的每一單位（ byte ）包含有 8 個 bits 。每一 BIT的表示非 0 卽 1 。byte 在記憶體中的位置就是它的地址，byte的地址和其內容可以寫成二進法，十進法，或十六進法（ hex ）來表示。例如

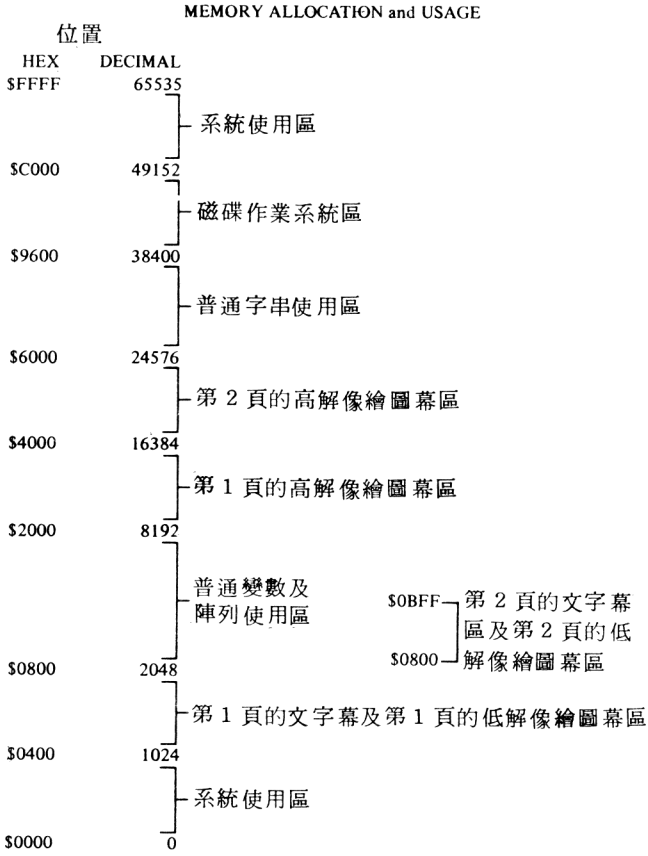


圖 2-1 記憶位置表

，如果記憶位置 5005 內容是 65 ，三種表示法是

0001001110001101	01000001	5005	65	138D	41
二進法		十進法		十六進法	

通常，我們是以十進法和十六進法來表示地址。雖然電腦的內部是二進法的運算，但它由於太煩瑣以致使我們表示上的不方便，所以只好取此捨彼了。

Apple Ⅱ的記憶位置是十進法中 0 到 65536 個地址。用十六進法表示是從 $ 0 到 $FFFF，當我們討論記憶位置如 $1B82(7042)時，通常指 $1B(27) 作“高位元組”而 $82(130) 作“低位元組”（ 7042 = 27 × 256 + 130 ）。

注意 1：雖然數目 1B 不太可能被誤解爲十進位數目，但是數目 82 就有可能被誤成十進位或十六進位數目。因此我們有必要區別是何進位法。因爲十進法是在我們語言中已確立了，所以我們大可不必去影響它；反倒是十六進法須特別標明，有三種方法常被使用：$1B_{16}$，1BH 或 $1B。而因 Apple 的所有手册是採第三種方法所以本書亦循例採之。

注意 2：附錄 1 更深入的討論十進法和十六進法。

記憶體位置 $400(1024) 到 $7FF(2047) 的使用有二：文字顯示和低解像繪圖。在TEXT 模式時，電腦會將這些記憶體地址的內容當作文字而顯示於 24 行，每行 40 個字元的位置。在GR 模式時，電腦會把記憶體位置 $400 到 $7FF 的內容當作繪圖與文字的混合，顯示幕底下四行顯示文字，其上爲 40 × 40 的繪圖矩陣。

記憶體地址 $800(2048) 到 $BFF(3071) 也可以作文字或繪圖的顯示，但此區域內存在有 Applesoft 的程式，所以技巧地把記憶體位置 $800 到 $BFF 當作文字或繪圖的第 2 頁，詳細情形在“第 2 文字頁”節中再來討論。

記憶體位置 $2000(8192) 到 $3FFF(16383) 是分配給第一頁的高解像繪圖使用。在HGR 模式時，電腦會顯示 280 × 160 的繪圖型及顯示幕最下四行的文字顯示。至於此四行文字正是第一文字頁的最後四行（ $400 − $7FF ）。

第 2 頁的高解像繪圖在記憶體地址 $4000(16384) 到 $5FFF(

24575 ）。當在 HGR2 模式時，電腦會把記憶體的區域當作 280 × 192 繪圖型且不顯示文字。

其實在第二頁的高解像繪圖裏顯示文字於顯示幕的底端是辦得到的。文字可寫在第 2 頁文字的最後四行（ $800 – $BFF ）。此外，也有在第一頁的高解像繪圖裏顯示 280 × 192繪圖型但顯示幕底端就不呈現文字了。同理，第一及第二頁的低解像繪圖都可或不可顯示文字。以上這些模式的技巧會在“軟性開關”一節中再來描述。

記憶體的定址顯示 （Memory Addressing For Display Purposes）

記憶體位址 $400 至 $7FF 雖然同時顯示文字與第一頁的低解像繪圖兩種用途。但事實上位置的結構並非我們想像的樣子。圖 2.2說明了文字頁時的記憶體位置型態。

圖 2.2所示所表示的標題（ label ）可由下之敍述：

```
10 TEXT: HOME: VTAB 6: PRINT TAB (18): "TEXT"
20 PRINT TAB(15): "ADDRESSING"
```

或

```
10 TEXT:HOME
20 FOR ADDR=1681 TO 1684
30 READ X: POKE ADDR,X
40 NEXT ADDR
50 FOR ADDR=1806 TO 1815
60 READ X: POKE ADDR,X
70 NEXT ADDR
80 DATA 212, 197, 216, 212
90 DATA 193, 196, 196, 210, 197, 211, 211, 201,
   206, 199
```

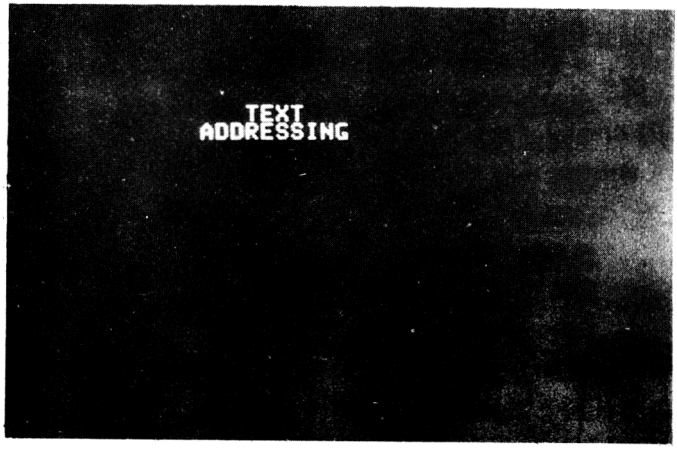


圖2.3所示爲以位置1024至2047 之記憶位置作低解像繪圖的情形。圖2.4則爲第一頁的高解像繪圖位置的情形。而第二頁的高解像繪圖頁亦有類似的結構。

使列位置（ row ）加上行位置（ column ）後所得的記憶體位置以符合每個文字的顯示幕位置。例如，字母“X”所佔區域的位置地址是列位地1664加上行位址19而得記憶體位址1683。字母“R”之記憶位址是列位址1792加上行位址17因此字母R的記憶地址1792 + 17 = 1809 。

欲得某一顯示幕位置的地址，讀者只需把對應的列位置與行位置相加就可。例如，圖2.3的陰影區的記憶位置，即爲列位置1664加上行位置25爲1689 。地址1689控制了兩塊區域（ block ）。上塊區的顏色由地址1689的低次半位元組（ lower nybble ）的內容決定；下塊區的顏色由其高次半位元組（ upper nybble ）的內容決定。

圖2.4所指的每塊區域均代表8個位元組。則列行相加之和的地址爲此塊區域內最頂端位元組的地址。緊接之後的7個位元組的地址乃每次加1024求得。

文字幕 （TEXT）

筆者用BASIC語言中POKE 命令來說明顯示幕位置結構。POKE

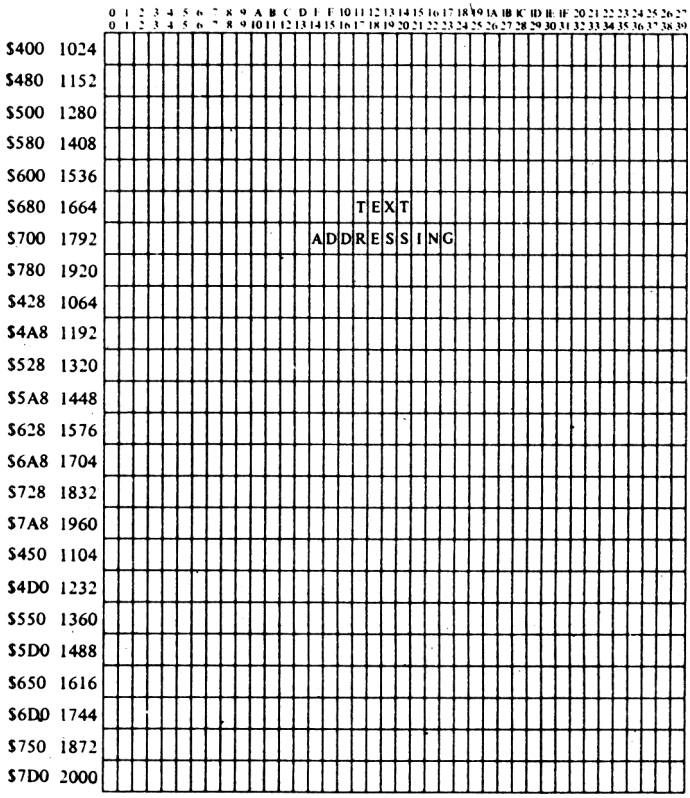

0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
$400 1024
$480 1152
$500 1280
$580 1408
$600 1536
$680 1664
$700 1792
$780 1920
$428 1064
$4A8 1192
$528 1320
$5A8 1448
$628 1576
$6A8 1704
$728 1832
$7A8 1960
$450 1104
$4D0 1232
$550 1360
$5D0 1488
$650 1616
$6D0 1744
$750 1872
$7D0 2000
TEXT
ADDRESSING

圖 2-2 文字幕位置結構

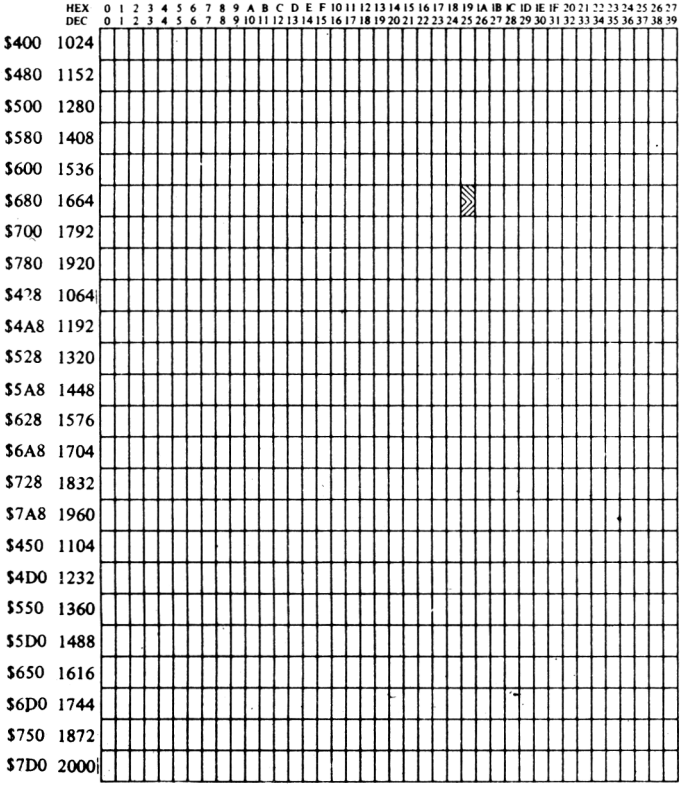
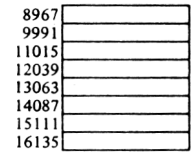

HEX 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27
DEC 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
$400 1024
$480 1152
$500 1280
$580 1408
$600 1536
$680 1664
$700 1792
$780 1920
$428 1064
$4A8 1192
$528 1320
$5A8 1448
$628 1576
$6A8 1704
$728 1832
$7A8 1960
$450 1104
$4D0 1232
$550 1360
$5D0 1488
$650 1616
$6D0 1744
$750 1872
$7D0 2000

圖 2-3　GR 幕位置結構

陰影區的八位元組地址



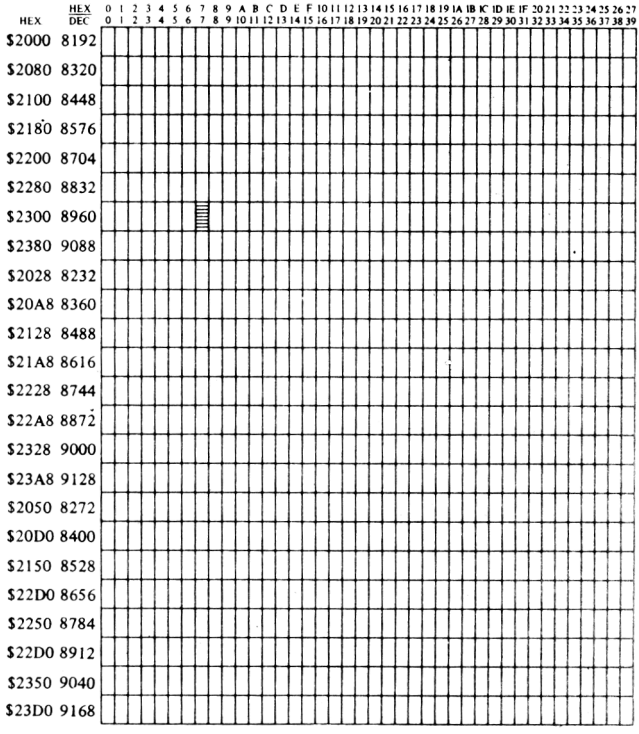


圖 2-4 MGR 幕位置結構

命令是每次將一內容存入某一記憶體位置。例如，POKE 1390,25 是指數目 25 存入地址 1390 內，若 Apple Ⅱ 在 TEXT 模式時，則此指令猶如將一個黑白反倒（白底黑字）的英文字母 Y，顯示在第 19 列的第 30 個位置上（原文是第 31 個位置，可能是錯誤的）。由於

25是黑白顛倒的Y電碼（請查閱Apple Ⅱ參考手册的ASCII字元碼），因此，顯示黑白反倒的Y。顯示Y的位置請參考文字顯示幕的位置結構（圖2.2）。讀者若想在黑白顛倒Y的正上方顯示一閃爍的Y（ASCII字元碼爲89），可執行POKE 1262,89指令。另POKE 1518,217則在黑白顛倒Y的正下方顯示一正常（黑底白字）的Y。

> 注意：欲連續執行上述情形，則顯示幕的席捲（scrolling）將改變Y所在的實際位置。不信，您可打入下之命令：
>
> ```
> TEXT: HOME: POKE 1390,25:POKE 1262,89:
> POKE 1518,217 (RETURN)
> ```

低解像繪圖幕　（GR）

在GR 模式時，記憶體位置的情況又完全不一樣了。此時顯示幕是40 × 40 的繪圖型且底端有四列文字幕。此記憶位置的內容控制兩個彼此重疊的圖形方塊區域的色彩，上塊區域的顏色由記憶位置的低次四位元內容決定；下塊區域的顏色由其高次四位元的內容決定。暫用下面的例子來說明：

因爲十進數25等於$19因此POKE 1390,25 產生一橘色（彩色碼9）的方塊重疊在橘紅色（彩色碼1）的方塊上。位置1390之記憶位置控制了（30,36）與（30,37)的顯示幕區域，因此（30,36）之方塊爲橘色，（30,37）之方塊爲紫紅色。

再如，十進數89等於$59，而位址1262 的記憶位置控制了方塊（30,34）與（30,35）的內容。因此POKE 1262,89 將（30,34）的方塊變成橘色，令（30,35）呈灰色。同理，POKE 1518,217 使（30,38）呈橘色，（30,39）呈黃色。

高解像繪圖幕（HGR）

在HGR與HGR2 模式時，所呈現的圖樣，幾乎是某一部份記憶位置中內容的逐一顯示。於此，我們以圖2.4所示的位置結構圖與POKE命令，來舉例說明。

在HGR模式時，顯示幕的最左上位置由$2000(8192)的記憶位置控制。其實，此記憶位置的前7個位元顯示HGR 顯示幕最頂列的前七個點。第1位元控制最左一點，第2位元控制第2點，依此類推。第8位元用於控制彩色，不顯示出來。（至於細節，請翻閱Apple參考手册）。

例如，假設位置8192的記憶位置內容18（POKE 8192,18即可使此位置含18）。18的二進制爲00010010，但因圖形顯示的順序（由左而右）正好與位元順序（由右而左）相反，所以第2點與第5點應該亮（ON），其餘點不亮（OFF）。

圖2-5 內容“18”顯示情形

舉例說明一種點圖型顯示的方式。

程式2·1：點圖型法

```
1  REM PROGRAM 2.1 (BIT PATTERN)
2  REM PLACES THE LETTER A ON THE HI RES SCREEN
10  DATA 8,20,34,34,62,34,34
20  HOME : HGR : VTAB 21
30  FOR I = 8192 TO 14366 STEP 1024
40  READ X
50  POKE I,X
60  INPUT K$
70  NEXT I
80  END
```

每次您一按RETURN鍵就會再增加一個這個字母的位元組。

行號30的FOR廻路分別指示了位址8192，9216，10240，11264，12288，13312與14366 等幾個記憶位置。程式把記憶位置內容顯現在顯示幕的最左上角。行號30至70廻路執行的結果，把這些記憶位置的內容顯示成：

記憶位置	點圖型（反次序）	
	十進制	二進制
8192	8	0001000
9216	20	0010100
10240	34	0100010
11264	34	0100010
12288	62	0111110
13312	34	0100010
14366	34	0100010

顯示字

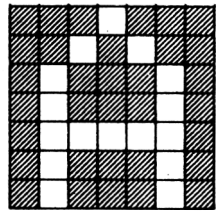


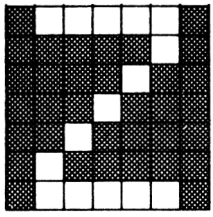
圖 2-6　點圖型顯示“A”

這些記憶位置的內容顯示後，就形成一個大寫的英文字母“A”。

以上之討論僅提供一種在繪圖幕上顯示出文字的方式，至於詳細的細節留待第3章再作討論。目前，僅向讀者先說明定義這字元的程序。

將讀者欲顯示的文字或符號繪成一張細方格表；7 × 7的格式。例如，英文字母“Z”的顯示如圖2.7所示。將每一點圖型轉換成一

顯示字



點圖型（反次序）	
二進制	十進制
0111110	62
0100000	32
0010000	16
0001000	8
0000100	4
0000010	2
0111110	62

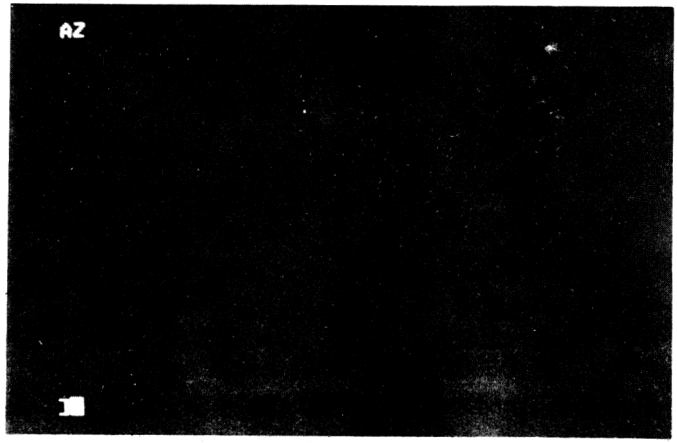
圖 2-7　點圖型顯示“Z”

個數目碼；然後將這些點圖型 POKE 到記憶體內。記憶位置可由圖 2.4 求知。

程式 2.2使圖形顯示幕最左上角上並肩地顯示出英文字母 “A” 與 “Z” 。

程式 2·2 ：點圖型字元

```
1   REM PROGRAM 2.2 (BIT PATTERN CHARACTERS)
2   REM PLACES LETTERS A AND Z ON THE HI RES
    SCREEN
10   HOME : HGR : VTAB 21
20   DATA 8,20,34,34,62,34,34
30   FOR I = 8192 TO 14366 STEP 1024      Letter A
40   READ X
50   POKE I,X
60   NEXT I
70   DATA 62,32,16,8,4,2,62
80   FOR I = 8193 TO 14367 STEP 1024      Letter Z
90   READ X
100   POKE I,X
110   NEXT I
120   END
```



注意與建議

設計一個程式，以繪圖顯示幕上顯示出一個簡短的英文單字，或讀者的英文名字的首字母。設計之前，得先用方格紙概略描繪欲顯示文字的圖樣。

彩　度（Colors）

筆者的意思並不是要强調高解像繪圖幕的彩色圖的顯示特色，而是有一個重要的限制是學習Apple Ⅱ者必須知曉的。此限制就是並非每一塊區域都能顯示出每一種顏色。執行程式 2.3，相信會有收穫的。

程式 2·3：彩度

```
1   REM PROGRAM 2.3 (COLOR PROBLEM)
2   REM PLOTTED LINES ARE NOT VISIBLE
10   HOME : HGR : VTAB 21
20   HCOLOR= 2
30   FOR I = 2 TO 5
40   HPLOT 7 * I,10 TO 7 * I,30
50   NEXT I
60   END
```

注意，雖然程式執行了四條垂直線，但在顯示時只能看到兩條。乃因 2 號色（藍）只可在偶數行畫出。除了黑色及白色外，其餘顏色均有類似的限制。因此，1 號和 5 號色也只能在奇數行顯示；2 號與 6 號色只能在偶數行呈現。而黑與白是任一行都顯示的。

爲了避免把垂直彩色線畫在無法呈現的位置上，向讀者介紹一種雙重繪圖技巧：若程式指令

```
HPLOT X,Y1 TO X,Y2
```

則加上以下指令

```
HPLOT X+1,Y1 TO X+1, Y2
```

程式2.4說明了另一個彩色問題。

程式2·4：混合彩度

```
1  REM PROGRAM 2.4 (PROBLEM WITH MIXED COLORS)
2  REM PLOTTING OVER BACKGROUND COLOR
10  HOME : VTAB 21: HGR : HCOLOR= 2
20  HPLOT 0,0
30  CALL 62454
40  HCOLOR= 5
50  HPLOT 1,1 TO 100,100
60  END
```

將顯示幕顏色令爲藍色。

APPLE Ⅱ上所有色彩的代號爲：

0—黑色1	4—黑色2
1—綠色（可變）	5—不一定
2—藍色（可變）	6—不一定
3—白色1	7—白色2

其中，1，2，5，6等顏色可變，視顯示幕的色彩控制調整情形而定。

故此，若欲使顯示的圖樣混合了很多種顏色時，那最好能選擇0，1，2，3等幾種色彩，或4，5，6，7等幾種色彩。程式2.4選定了藍色（行號10），然後以之作底色（行號20，30）。在以5號色畫對角線時（行號50），色彩的混合效果若非產生一種非常令人迷惑的圖樣，就產生一種令人看起來非常舒服的圖樣。而其感受就得看讀者的玩味了。

解像度 （Resoloution）

“解像度”是指繪圖所用繪圖面上的方格數。在Apple Ⅱ中，解像度可用HPLOT X,Y命令來控制。其中，X爲0至279 間之任意整數值，Y爲0至191間之任意整數值。因此，我們說高解像繪圖幕的解像度爲280 × 192 ，而正好產生53760個方塊或點。

但，要進一步的澄清。因爲，可能有人一開始都會發現，這些方

塊區域並非每一塊都能畫出每一種色彩。例如

```
HGR: HCOLOR=1: HPLOT 20,10 to 20,100
```

執行這指令後，您會發現，垂直線並未呈現。事實上，解像度是 140 × 192 ，而非 280 × 192 。

有許多黑白線的實際解像度也只可 140 × 192 ，例如：

```
HGR: HCOLOR=3: HPLOT 10,10 to 100,100
```

原來想繪的線應是“薄薄”的，但事實上却大異其趣。不要清除顯示幕，試試

```
HCOLOR=0: HPLOT 9,10 TO 99,100
```

將發現，原來前面的一條線比現在這一條還要“寬”。

但就另一角度而言，黑白圖片獲得了較佳的解像度。

程式2.5舉例如何達成更高的解像度。

程式 2·5：更高解像度

```
1  REM PROGRAM 2.5 (HIGHER RESOLUTION)
2  REM DISPLAYS 14 DOTS PER BYTE
10  HOME : HGR
20  FOR I = 1 TO 14
30  READ X
40  POKE 14800,X
50  VTAB 22: PRINT I
60  FOR J = 1 TO 500: NEXT J
70  NEXT I
80  DATA 1,129,2,130,4,132,8,136,16,144,32,160,
    64,192
```

注意，資料成對出現，彼此差128各位元的點圖型只有高次位元的不同。

注意程式的執行，讀者將看出改變一個位元組時所顯示出的連續十四個點。而因繪圖顯示幕寬爲 40 個位元組，故實際上其解像度如同爲 560 × 192 。

上面的討論重點是，Apple Ⅱ的高解像度並不是永遠固定的，而 280 × 192 的解像度也不是最理想的，有時讀者若欲獲得更高的解像顯然必須再經過程式設計的努力。而這努力已超過本書的重點。有興趣的讀者可進一步參考其它有關的書籍。

直線的定義 （What Is A Line？）

有時讀者不難體會，字典裡的定義經常無法應用至日常生活的實際狀況。當我們將字典裡對“直線”之解釋，應用至繪圖幕上所顯示的圖樣時，您的發現又是如何呢？理論上，直線的寬度（即厚度）應爲零。以繪圖顯示幕畫出直線時，顯示幕上的點非亮（ON）即暗（OFF）。由於點的大小不爲0。故其表示的線寬度亦不爲零。此外畫點是依列與行的方向排列的，故以繪圖幕所畫出的對角線看起來亦無法完全平直。

如圖2.8之直線。若我們使直線通過每一個畫點發亮，則所得的直線圖像將會出現階梯的痕跡。

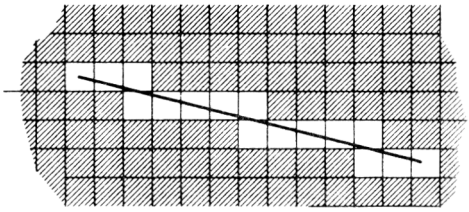


圖 2-8 顯示幕的直線

Applesoft 的HPLOT X1,Y1 TO X2,Y2命令所畫出的“直線”，亦是像圖2.8示的樣子。會出現階梯突角之痕跡。

直線定義的方式亦會影響到圖像。怎麼說呢？欲求出直線時，Applesoft必須決定那一個顯示幕畫點必須點亮。由於使用計算的結果，直線的方向是非常重要。例如：

```
HGR: HCOLOR=3:HPLOT 10,10 TO 100,90
```

接著試：不要清除前之敍述

```
HPLOT 100,90 TO 10,10
```

仔細看，雖然要畫同一條線，但圖像却略有差異。這完全是因爲直線定義的方向倒反的緣故。再試下面方式：

```
HGR: HCOLOR=3: HPLOT 10,10 TO 100,90
HCOLOR=0: HPLOT 100,90 TO 10,10
```

將發現，第二條線（黑色）呈現，但第一條線（白色）被塗掉，但留下了少許的痕跡。

雖因直線的方向定義的不同所造成的差異通常並不很重要。但，有時注意定義的細節，往往會使圖像比較滿意。

HCOLOR 設定（Hcolor Settings）

通常使用繪圖命令之前，必須先指明繪圖所用的記憶區域（HGR或HGR2），以及顏色的選擇（HCOLOR = C 其中 C =0，1，2，3，4，5，6或7）。HCOLOR一經決定後，直到下一次“HCOLOR=”敍述出現前，其顏色固定不變，即令執行過了HPLOT與HPLOT TO亦然。不過，只要是指明HCOLOR，則緊接的應爲HPLOT命令，而不能是HPLOT TO的命令。原因是：執行HPLOT TO X,Y 敍述時，指令所用的將是最後一次（繪圖時）用過的顏色。但在執行HPLOT X,Y命令時，其恒爲使用最新的“HCOLOR=”敍述所設定的顏色。

有時爲了方便，在好幾個HPLOT TO X,Y 敍述之間更換顏色，而不需再插入HPLOT X,Y命令，以重新設定HCOLOR，我們可藉用下面的方法“騙過”標準的Applesoft程序，來達成目的。

地址228（$E4）的記憶位置內容有最新指明的HCOLOR的色碼。而剛剛畫圖所用過的色碼則存在28（$1C）的記憶位置內。因此，只要我們“偸偸”改變位址28之記憶位置的內容（用POKE），我們卽可“騙過”Applesoft，令其改變顏色。

試試以下程式：

程式2·6：HCOLOR設定

```
1  REM PROGRAM 2.6 (HCOLOR SETTINGS)
2  REM CONVENTIONAL COLOR CONTROL
10  HOME : VTAB 21: HGR : HCOLOR= 2
20  HPLOT 20,20 TO 50,75
30  HCOLOR= 3
40  HPLOT TO 100,10
50  HPLOT 10,100 TO 200,150
60  END
```

注意，行號 30 雖然一開始就指定一新的顏色。但事實上一直到行號50才產生效用。再試試下面程式：

程式 2·7 ：HCOLOR 控制的改良

```
1  REM PROGRAM 2.7 (IMPROVED HCOLOR CONTROL)
2  REM SETTING HCOLOR WITH POKES
10  HOME : VTAB 21: HGR : HCOLOR= 2
20  HPLOT 20,20 TO 50,75
30  POKE 28,127: REM COLOR CODE OF HCOLOR=3
40  HPLOT TO 100,10
50  HPLOT 10,100 TO 200,150
```

注意：行號30的顏色更換，但在行號40開始發生作用（ Apple-soft 認爲剛剛畫的HCOLOR 爲 3 ），不過行號 50 又將色彩變回最近的HCOLOR= 敍述時所指明的顏色。

使用此程序時，讀者手邊必須有HCOLOR的彩色碼：

HCOLOR	彩色碼	色 彩
0	0	黑
1	42	綠
2	85	藍
3	127	白 1
4	128	黑 2
5	170	不 定
6	213	不 定
7	255	白 2

記得第一章的程式1.8曾以一系列值改變繪圖的顏色，當時的技巧就是這裏所討論的東西。

軟性開關（Soft Switches）

程式中，HGR 命令有好幾種功用。包括：

1. 顯示第 1 頁的高解像繪圖頁。
2. 將第 1 頁的高解像繪圖頁清除成黑色。
3. 將第 1 頁的高解像繪圖頁看成是後面繪圖要用的區域。

有時，您可能欲顯示某一繪圖頁的內容，而不準備除去，或輪流交換顯示兩繪圖頁。藉此效果，尤其是動態的顯現。我們可以在畫第 2 頁時顯示出第 1 頁，然後在畫第 1 頁時呈現第 2 頁，如此輪流交換。而單靠 HGR 與 HGR2 命令是無法滿足以上要求。

因此控制就須經由“軟性開關”來達成。此種技巧是以軟體方式存取八個記憶位置。這八個位置可以看成是下面四個開關

1. 文字／繪圖開關
2. 全顯示幕／混合式顯示幕開關
3. 第 1 頁／第 2 頁開關
4. 高解像／低解像開關

每一開關均只控制以上四種的顯示方式。有好幾個 BASIC 命令可一次同時控制數個開關。例如：HGR 命令將 1 號開關設定爲繪圖型，2 號開關設定成混合式顯示幕，3 號則設定成第 1 頁，4 號開關設定

表 2-1　軟性開關

開關	控制	BASIC 命令
1	繪圖幕	POKE −16304,0
	文字幕	POKE −16303,0
2	全文字或繪圖顯示幕	POKE −16302,0
	混合文字和繪圖顯示幕	POKE −16301,0
3	第一頁	POKE −16300,0
	第二頁	POKE −16299,0
4	低解像度	POKE −16298,0
	高解像度	POKE −16297,0

成高解像。GR 命令略同，而將 4 號開關設定成低解像。（注意，指命除了設定開關外，尚做一些其它事呢）。

若讀者只欲個別控制每一開關時，可以利用 POKE 命令。控制法如表 2.1 。

POKE 命令可將每一開關設定在其中一個位置。而設定之後，除非更動，否則，固定不變。

因爲四個開關都能夠各個控制，看起來似乎有 16 種可能顯式型態，其實不然。例如，若第 1 個開關設定在文字位置上（ POKE−16303,0 ），則 2 至 4 號開關設定將完全無效。

因此表 2.2所示，只有十種看得見的不同顯示型態。

表 2-2 十種不同的軟性開關顯示型態

型態	BAS IC 命令
文字第 1 頁	POKE −16303,0: POKE −16300,0
文字第 2 頁	POKE −16303,0: POKE −16299,0
全顯示幕第 1 頁 低解像度繪圖幕	POKE −16298,0: POKE −16304,0 POKE −16300,0: POKE −16302,0
混合顯示幕第 1 頁 低解像度繪圖幕	POKE −16298,0: POKE −16304,0 POKE −16300,0: POKE −16301,0
全顯示幕第 2 頁 低解像度繪圖幕	POKE −16298,0: POKE −16304,0 POKE −16299,0: POKE −16302,0
混合顯示幕第 2 頁 低解像度繪圖幕	POKE −16298,0: POKE −16304,0 POKE −16299,0: POKE −16301,0
全顯示幕第 1 頁 高解像度繪圖幕	POKE −16297,0: POKE −16304,0 POKE −16300,0: POKE −16302,0
混合顯示幕第 1 頁 高解像度繪圖幕	POKE −16297,0: POKE −16304,0 POKE −16300,0: POKE −16301,0
全顯示幕第 2 頁 高解像度繪圖幕	POKE −16297,0: POKE −16304,0 POKE −16299,0: POKE −16302,0
混合顯示幕第 2 頁 高解像繪圖幕	POKE −16297,0: POKE −16304,0 POKE −16299,0: POKE −16301,0

所以欲從一種顯示型態轉換成另一種顯示型態時，讀者只需改變開關即可，例如，想從全顯示幕第 1 頁的高解像繪圖型態轉換成全顯示幕第 2 頁高解像繪圖型態時，只需變換 3 號開關（ POKE −16299 ,0 ）即可達成。同理，想從全顯示幕高解像繪圖第 1 頁轉換成混合

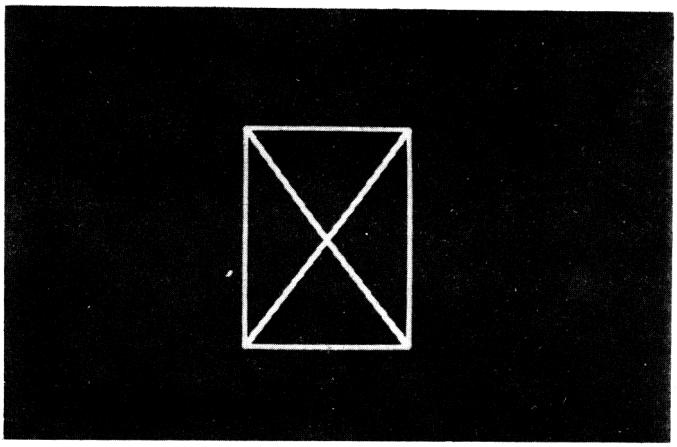
顯示幕高解像繪圖第1頁時，只需變換2號開關（ POKE－16301，0 ）即是。

程式2.8舉例了以軟性開關選定其中一種型態。

程式2·8：閃爍X及其方塊

```
1 REM PROGRAM 2.8 (FLASHING X AND SQUARE)
2 REM ILLUSTRATES USE OF SOFT SWITCHES
10 HGR : HCOLOR= 3
20 HPLOT 90,40 TO 190,40 TO 190,160 TO 90,160 TO
   90,40
30 HGR2
40 HPLOT 90,40 TO 190,160: HPLOT 190,40
   TO 90,160
50 FOR I = 1 TO 100
60 FOR J = 1 TO 50: NEXT J
70 A = 1 - A
80 POKE -16299 - A,0
90 NEXT I
```

設定顯示開關。



此程式使能將兩個圖形頁中任一頁之圖像顯示，但却不能輕易地修改圖樣。但若利用顯示幕顯示文句或另一個圖形頁時，改變圖形頁

上的圖像，將有令人滿意的效果。軟性開關雖然控制顯示型態，但對用以繪圖的圖形頁却無作用。

究竟是由那一個繪圖頁來繪圖則由地址$E6(230)的記憶內容決定。使用第1頁，則此位置內容$20(32)；使用第2頁，則此位置內容$40(64)。由于位址E6(230)之記憶位置內容可單獨控制，不受型態的牽制，因此，可以一面顯示某頁，而另一面準備顯示另一頁。

程式2.9說明了顯示的同時另一頁準備顯示的方法。

程式2.9產生了一個先慢慢放大，然後再逐漸縮小的方形。放大與縮小的變換將一直執行至CTRL-C或RESET鍵被按中斷而止。

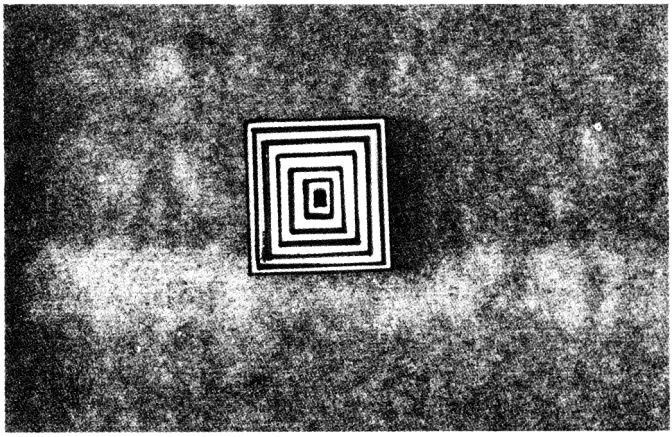
程式內，行號50 設定了繪圖頁。注意，繪圖的動作永遠發生於不是正在顯示的頁上，60至80行號則除去原先的方形繪圖(只需畫

程式2·9：動態方塊

```
1   REM PROGRAM 2.9 (ANIMATED SQUARES)
2   REM ILLUSTRATES ANIMATION TECHNIQUES
10   HGR : HGR2
20 A = 1:B = -1:C = 2:D = 1
30   FOR I = 1 TO 20
40   FOR J = 1 TO 2
50   POKE 230,32 * J
60 A = A + B
70   HCOLOR= 0
80   HPLOT 140 - A,96 - A TO 140 + A,96 - A
     TO 140 + A,96 + A TO 140 - A,96 + A
     TO 140 - A,96 - A
90 A = A + C
100   HCOLOR= 3
110   HPLOT 140 - A,96 - A TO 140 + A,96 - A
      TO 140 + A,96 + A TO 140 - A,96 + A
      TO 140 - A,96 - A
120   POKE - 16299 - D,0
130 D = 1 - D
140   NEXT J,I
160 B = -B:C = -C
170   GOTO 30
```

設定繪圖頁

設定開關



成背景色卽是）。行號90至110 則畫出一個更大或更小的方形。行號120乃一交替變換顯示開關，以顯示出剛畫好的方形。程式繼續進行，除去未顯示的圖樣，並重畫另一顯示的圖樣。

記憶體搬運（Memory Move）

通常，資料常常必須由某一記憶體搬運到另一記憶體。不過，此處所要討論的主題主要是有關圖形方面的探討。由於 Apple Ⅱ的繪圖實際上只是顯示某記憶體的內容。因此，搬運記憶體的內容事實上就是搬動圖樣的資料。

> 注意：雖然通常都講“搬動”（ move ），但事實上“抄錄”（ copy ），因爲，搬動作業產生後，被搬動的原始資料仍然保持不變。

將資料由記憶體間搬運的方法之一，卽利用BASIC 命令中的PEEK與POKE。PEEK讀取某一記憶位置的內容（讀取後，原來記

憶位置的內容不變）。而POKE是將新值存入某一記憶位置內（POKE改變了原先記憶位置的內容），舉例：命令

```
POKE 9, PEEK (8)
```

即是讀取位址8之記憶內容（ PEEK （8））然後存入位置9的記憶地址內。執行後，地址8之記憶位置的內容保持不變。即記憶地址8及9現在內容是一樣的。

利用PEEK和POKE我們即可將記憶體間的內容相互抄錄。例如：將第2頁的高解像繪圖頁的內容抄錄至第1頁的高解像繪圖頁。

```
110 FOR I = 8192 TO 16383
120 POKE I, PEEK(8192 + I)
130 NEXT I
.
.
.
```

此法的缺點是，速度太慢，所幸尚可利用機器語言層次的程式來執行。而只須事先做下列的事前設定，則已知的BASIC 程式就可利用MOVE程式來執行了

```
POKE 768,160: POKE 769,0
POKE 770,76: POKE 771,44
POKE 772,254
```

經過了此些設定後，欲將記憶體的內容抄錄時，程式即需指出：

1. 來源記憶體 SOURCE的起始位址。
2. SOURCE的最終位置。
3. 目的記憶體 DESTINATION的起始位址。

指定以後，CALL 768即可叫用MOVE 程式3，自動地完成搬運作業。

值得說明的，上列三個指明的每一個地址都必須以兩個階級來指定。爲了使MOVE程式正確執行，兩個階級的高次與低次位元組必須事先分別如下所示地存好：

位置地址	所含內容的位元組
60	SOURCE之起始位址的低次位元組
61	SOURCE之起始位址的高次位元組
62	SOURCE之最終位址的低次位元組
63	SOURCE之最終位址的高次位元組
66	DESTINATION之起始位址的低次位元組
67	DESTINATION之起始位址的高次位元組

例如，程式2.10即設定了MOVE 程式所必須用的初值，然後叫用MOVE程式，將第2頁高解像繪圖頁的內容抄錄到第1頁的高解像繪圖頁中，然後再以地址$6000至$7FFF的記憶位置內容抄錄到第2高解像繪圖內。

程式2·10：記憶體搬運的副程式

```
1   REM PROGRAM 2.10 (MEMORY MOVE SUBROUTINE)
2   REM NOT A DISPLAY PROGRAM
10   REM INITIALIZE
20   POKE 768,160: POKE 769,0
30   POKE 770,76: POKE 771,44
40   POKE 772,254
50   REM PAGE 2 TO PAGE 1
60   POKE 60,0: POKE 61,64
70   POKE 62,255: POKE 63,95
80   POKE 66,0: POKE 67,32
90   CALL 768
100   REM PAGE 3 TO PAGE 2
110   POKE 60,0: POKE 61,96
120   POKE 62,255: POKE 63,127
130   POKE 66,0: POKE 67,64
140   CALL 768
```

使用PEEK和POKE的方法來抄錄記憶體內的內容，若用BASIC程式來設計或許會較簡化。不過，有關MOVE程式之初值設定及被用，用機器語言來執行速度會快一點。

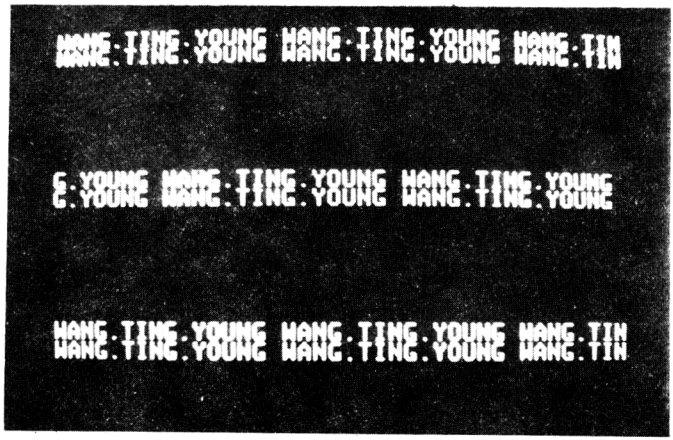
使用MOVE程式必須注意一點：若資料搬運的目的區與來源區相重疊，則很可能發生來源區中有部份資料在尚未被抄錄前即因先被寫

入而遭破壞的情形。這種情形一定要避免，但是，有時候設計者會技巧地故意這樣做，爲了反複複製某圖案或其它目的。程式 2.11 即將輸入一個姓名，將之印在一清徹顯示幕的最頂端，然後再將之抄錄至頁上的其餘部份。

程式 2·11：姓名複印

```
1   REM PROGRAM 2.11 (NAME COPIER)
2   REM USES MEMORY MOVE TO DUPLICATE A STRING
10   POKE 768,160: POKE 769,0
20   POKE 770,76: POKE 771,44
30   POKE 772,254
40   TEXT : HOME
50   INPUT "YOUR NAME, PLEASE";A$
60   HOME : PRINT A$
70  L = LEN (A$) + 1
80   POKE 60,0: POKE 61,4
90   POKE 62,255 - L: POKE 63,7
100   POKE 66,L: POKE 67,4
110   CALL 768: WAIT -16384,128
120   HOME
```

暫停直到有按鍵被壓爲止；然後清除顯示幕。



其它繪圖頁（Additional Graphics Pages）

程式2.10時我們一起討論過，把位址$6000至$7FFF記憶位置的內容抄錄至第2高解像繪圖頁內，其實，除非地址$6000至$7FFF等記憶位置上原來就儲存有一張高解像的圖樣，否則，先前的抄錄動作是沒有意義的。倒是讀者欲將一張圖案存入位址$6000至$7FFF記憶位置時，筆者可提供三種方法：

1. 自磁碟讀取（由讀者向美國郵購的磁碟命名爲：BLOAD PICTURE，A$6000）；
2. 先畫在第1或第2高解像繪圖頁，然後再抄錄至位址$6000至$7FFF之記憶位置。
3. 直接畫在“第3高解像繪圖頁”上，（$6000－$7FFF）。

而以第3種的選擇以BASIC敍述最易達成，程式2.12即是用此法。選此法，必須先指出“第3頁”的繪圖記憶區（POKE 230，96），將顯示幕清除成黑色（CALL 62450），然後再以BASIC繪圖命令開始繪圖。而既然討論到增加繪圖頁數，何不使用“第4頁”呢？事實上，位址$8000至$9FFF等記憶位置亦可繪圖。不過，有一點要小心，此記憶頁有部份記憶區被磁碟作業系統（DOS）佔用。因此在您清除“顯示幕”或在此記憶頁繪圖時，有部份的DOS將被破壞而無法動作。因此，程式行號100特別令DOS暫時中斷，等至執行完後再使DOS接上系統。

高解像繪圖“第5頁”（$A000－$BFFF）的使用與“第4頁”相同。

使用“第四頁”或“第五頁”時，避免存有字串。由於HIMEM（恰在DOS下面）底下經常所存的就是字串，因此，解決方法是，在程式開始就先設定HIMEM使之成爲8191。如此就可以使字串正好儲存在第1高解像繪圖頁下面。如

```
1 HIMEM: 8191
```

程式2·12：其他繪圖頁

```
1   REM PROGRAM 2.12 (ADDITIONAL GRAPHICS PAGES)
2   REM DRAWS, DISPLAYS 4 PAGES OF GRAPHICS
10   TEXT : HOME : HCOLOR= 3
20   POKE 768,160: POKE 769,0
30   POKE 770,76: POKE 771,44
40   POKE 772,254
50   POKE 230,64: CALL 62450
60   HPLOT 140,10 TO 140,180

70   POKE 230,96: CALL 62450
80   HPLOT 120,10 TO 120,180
90   HPLOT 160,10 TO 160,180
100   IN# 0: PR# 0
110   POKE 230,128: CALL 62450
120   HPLOT 110,10 TO 110,180
130   HPLOT 140,10 TO 140,180
140   HPLOT 170,10 TO 170,180
150   POKE 230,160: CALL 62450
160   HPLOT 110,10 TO 110,180
170   HPLOT 170,100 TO 10,100 TO 110,10
180   PRINT "THIS PROGRAM WILL DISPLAY ONE OF
      FOUR": PRINT
190   PRINT "HIGH RESOLUTION IMAGES": PRINT
200   PRINT "PRESS KEYS 1, 2, 3 OR 4 TO SEE
      THEM": PRINT
210   PRINT "PRESS 'ESC' TO END THE PROGRAM":
      PRINT
220   PRINT "PRESS 'RETURN' TO RETURN TO THIS
      MENU"
230   POKE -16297,0: POKE -16302,0
240   VTAB 18: GET A$: PRINT :X= FRE (0)
250   IF ASC (A$) = 27 THEN TEXT : HOME :
      PRINT "DOS IS DEAD!": END
260   IF ASC (A$) = 13 THEN POKE -16303,0:
      GOTO 240
270 A=VAL (A$): ON A GOTO 290,340,390,440
280   GOTO 240
290   POKE -16304,0
300   POKE 60,0: POKE 61,64
310   POKE 62,255: POKE 63,95
320   POKE 66,0: POKE 67,32
330   CALL 768: GOTO 240
340   POKE -16304,0
350   POKE 60,0: POKE 61,96
360   POKE 62,255: POKE 63,127
370   POKE 66,0: POKE 67,32
380   CALL 768: GOTO 240
390   POKE -16304,0
```

清除230位置所指的顯示幕，然後畫上圖案。

清除230位置所指的顯示幕然後畫上圖案。

將第2，3，4，5頁移至第1頁。

```
400   POKE 60,0: POKE 61,128
410   POKE 62,255: POKE 63,159
420   POKE 66,0: POKE 67,32
430   CALL 768: GOTO 240
440   POKE -16304,0
450   POKE 60,0: POKE 61,160
460   POKE 62,255: POKE 63,191
470   POKE 66,0: POKE 67,32
480   CALL 768: GOTO 240
```

程式 2.12 舉例說明了軟性開關與第 1 及第 2 頁以外的記憶區高解像繪圖頁之能力的使用。當然，除非將之移至第 1 頁或第 2 頁上；否則，畫在“第 3 ，4 ，5 頁”上的圖形是看不見的。

由於須令DOS 不動作，故程式行號 250 處形成一個虛出口（ cold exit ）。因此行號 250 換成

```
250 IF ASC(A$) = 27 THEN TEXT: CALL 50688
```

則控制離開程式後，DOS自動接上，不過這個動作的先決條件是：磁碟控制器是已插在 6 號槽了（ slot 6 ）。

第2文字頁 （Page 2 Of Text）

Apple Ⅱ也提供顯示第 2 個文字頁（ POKE − 16303,0 ：POKE − 16299,0 ），但寫入此文句頁是困難的。雖然我們主要的興趣是高解像繪圖，但顯示第 2 文字頁的能力亦是值得討論的，特別是在第 2 高解像繪圖頁混合文字與圖樣中，尤須畫出第 2 文字頁的文字部份。

第 2 文字頁使用時面臨的第一個問題，文字頁佔用的記憶地址（ $800 ～ $BFF ）平常用於儲存 Applesoft BASIC程式用的，因此能使用第 2 文字頁，程式必須另謀它處來儲存，這可容易達成！因爲，Applesoft 程式之起始位置分別於位址 $67(103) 與 $68(104) 的記憶位置內，前者存低位元組，後者存高位元組。換言之，地址 103 的位置通常含內容 $01(1) ，及地址 104 則常含內容 $08(8)。（雖然程式記憶區自位址 $800 的位置留起，但此位置內容通常都含

零，因此，程式實際均由位址$801的記憶位置開始存起。）

因此，若BASIC 程式欲使用第1或第2文字頁以及第1或第2高解像繪圖頁時，則可以儲存程式的較大記憶區將由地址$6000(24576）開始。若我們希望將程式存入這個區域，則我們必須將位址103與104之記憶位置的內容分別設定成$01(1）與$60(96）。（196 × 256 = 24576 ）同時，亦須將程式開頭（ 0 ）存入位址$6000（24576） 的記憶位置內。欲使程式打入這些位置。我們可以先打入：

```
POKE 103,1
POKE 104,96
POKE 24576,0
NEW
```

然後可以打新程式了：

欲讀取且執行這個位置上的程式時，我們須打：

```
POKE 103,1
POKE 104,96
POKE 24576,0
RUN PROGRAM
```

以上四列可當成Applesoft 程式的一部份由鍵盤打入或執行EXEC文字檔案中得來。

知道如何將一個BASIC 程式存於第2文字頁以外的記憶區域後，接著我們來看看如何寫入這個文字頁的問題。其方法有二：

1. 先將程式印在第1文字頁，然後再以機器語言層次的MOVE程式，將第1文字頁的內容抄錄第2文句頁。
2. 直接印出在第2文字頁上。

但使用第2種方法時，程式必先指明第2頁作爲印出區（ printing）。通常，位址$29(41） 的記憶位置都會含一個4與7間的數目，指出位址$0400至$07FF之1024 個記憶位置，作爲儲存文字或低解像圖形的部份記憶區。若在位址$29之記憶位置存入8，9，10 或

11，則此位置就會指至位址$800至$BFF 的記憶位置。此時，印出的東西就會緊接印在第2頁上。不幸的是，每次碰到 RETURN 鍵（BASIC程式之每一敍述的結束）時，地址$29 之記憶位置的內容就會自動重新設定一介於4與7間的數值。因此，在每次欲印在第2頁前，位址$29之記憶位置的內容都必須重新設定一次。程式2.13即用此法。

程式2·13：兩個文字頁

```
1   REM PROGRAM 2.13 (TWO PAGES OF TEXT)
2   REM ILLUSTRATES PRINTING, DISPLAYING PAGE TWO
    OF TEXT
3   REM BE SURE TO SET PROGRAM POINTERS
4   REM AT LOCATIONS 103,104
5   REM BEFORE LOADING OR TYPING THIS PROGRAM
10   HOME : VTAB 10
20   FOR I = 2048 TO 3071: POKE I,160: NEXT I
30   POKE -16299,0
40   SPEED= 55
50   POKE 41, PEEK (41) + 4: PRINT "WE ARE NOW
     PRINTING ON PAGE 2."
60   POKE 41, PEEK (41) + 4: PRINT "GET READY FOR
     A CHANGE TO PAGE 1."
70   FOR I = 1 TO 1000: NEXT I
80   POKE -16300,0: VTAB 10
90   PRINT "WE ARE NOW PRINTING ON PAGE 1."
100   PRINT "WE WILL DISPLAY EACH OF THE TWO"
110   PRINT "TEXT PAGES REPEATEDLY"
120   PRINT "WHILE COUNTING FROM 1 TO 40"
130   SPEED= 255
140   FOR J = 1 TO 20
150   POKE -16299,0
160   VTAB 18: POKE 41, PEEK (41) + 4:
      PRINT 2 * J - 1
170   FOR I = 1 TO 1000: NEXT I
180   VTAB 18: PRINT 2 * J
190   POKE -16300,0
200   FOR I = 1 TO 1000: NEXT I
210   NEXT J
```

在設定指標(41)後立即印出。

黑白反倒圖像（Negative Images）

在前面說過，Apple所顯示的高解像圖形，事實上就是顯示某一段記憶位置的內容。在顯示的過程中，若某一記憶位元的值爲1，則這個位元所相對應的顯示點卽亮（ ON ）；反之，若某一記憶位元之值爲0，則其所相對應的顯示點卽暗（ OFF ）。

因此，若高解像繪圖記憶位置的內容若能做有系統的改變，則顯示幕的顯示圖樣亦將隨著改變。例如：將圖像記憶的每個記憶位元的值均反態卽1變0，0變1，則顯示的圖像亦勢變成黑白反倒。在第1頁高解像繪圖頁上，這可以下面的命令來完成：

```
FOR I = 8192 TO 16383: POKE I, 255 - PEEK(I):
NEXT I
```

而第2頁高解像繪圖頁則可以用下之命令來完成：

```
FOR I = 16384 TO 24575: POKE I, 255 - PEEK(I):
NEXT I
```

程式設計助手 （ Aids ）

讀取鍵盤

具有對談特色的程式均以 INPUT 或 GET 命令接受使用者的輸入。執行到二者命令其中一個時，計算機會自動停下來等待使用者的輸入，然後再繼續進行程式其它部份或顯示結果。有時候，我們會比較喜歡令程式一直執行至我們按某一個鍵，如 ESC 爲止。程式1.4卽是此例。在等待使用者輸入時，程式並未使圖形顯示暫時中止，而是連續不斷地一直檢查位址－16384的記憶位置（ PEEK(－16384)）。這個位置專門負責接收按鍵輸入。若您按 ESC 鍵〔 PEEK(－16384 ）＝155 〕，程式執行卽終止（行號120 ）。否則，程式一直執行。（有關 ASCII 的鍵碼，請參閱Apple 參考手册）

行號120另外還做一件事，當讀者在鍵盤上按一個鍵時，此按鍵的鍵碼卽存入－16384的記憶位置，並一直留在那兒。爲了能接受新

的按鍵，因此在讀取按鍵後，POKE－16368，0 命令順便將“鍵盤獲取信號清除零”。

程式1.9則提供另一個不以INPUT或GET命令做鍵盤輸入的例子。在程式中行號70讀取鍵盤按鍵；行號80至240分析按鍵，然後按鍵結果再控制程式的執行。此即程式1.9動作所採用的另一種方式。

程式冗長

若程式在執行中，點字串開始蔓延至圖形顯示幕，或程式的末端部份突然消失了，那就很可能是程式設計太長了。點字串的出現就代表程式變數已被存到用作高解像圖形顯示的部份記憶區。若程式太長而在$800至$1FFF 的現有記憶空間內容納不下，則程式內的HGR命令即會除去程式的末尾部份。

避免發生這種問題的方法很簡單。只要打入程式的過程中，或程式自磁碟取入後，以

```
PRINT PEEK(175) + 256 * PEEK(176)
```

敍述印出Applesoft程式之最後一個命令所在的記憶位置的地址，即可避免發生上述的現象。爲了能存得下程式內所定義的變數，這個印出位址的值必須小於8192（第1頁高解像繪圖頁的起始位址）。若發現空間不足，請採取下列的預防措施：

1. 將變數存在繪圖頁的上方。令LOMEM等於24576（將變數存在第2頁高解像繪圖頁之上）或16384（將變數存在第1頁高解像繪圖頁之上，但在第2頁之內）。LOMEM的值必須在程式一開始時先設定，如

```
1 LOMEM:24576
```

2. 將程式與變數均儲存在圖形頁上。這一部份的記憶體經常會不用。用以儲存程式的記憶區通常在位址2048（$800）至8191（$1FFF）之間。這裡大約有6000 個位元組之記憶空間。不過

，在第2繪圖頁與DOS間，在48K RAM的系統近乎有14000個位元組的記憶空間。較長的程式可以儲存在這裏。至于儲存的手續與“第2文字頁”一節所敍者相同，望讀者自行參考。

注意與建議

1. 有關Apple的記憶空間的使用，Apple Ⅱ 參考手册上已提供了許多很有價值的參考資料。加上最近剛出版的一些書刋，像Apple Orchard，MICRO與Nibble，上面也有介紹另Apple Orchard之Vol. 1，No. 2與Vol. 2，No. 1上均有專門討論有關高解像繪圖的文章。至於有關記憶位置結構方面，W.F. Luebbert.所寫的“What's Where in the Apple”就是--本最好的研讀資料。

2. 修改程式1.9，使其能執行：

S（將目前的圖像儲存在第2頁的高解像繪圖頁裏）使用記憶體搬運副程式。

T（顯示出第2高解像繪圖頁的圖樣）。以POKE作軟性開關。

O（顯示出第1高解像繪圖頁的圖樣）。以POKE作軟性開關選定。

R（再將第2頁上的圖像抄錄於第1頁）。使用記憶體搬運的副程式。

S，T，O與R每一個都應能自動轉移至C（浮標）模式。

在熟悉以上按鍵後，程式達成：

⑴將尚未完全畫好的圖像儲存在第2頁裏，然後繼續畫第1頁；

⑵將更進一步的繪圖（在第1頁上）與先前的繪圖（第2頁上）相比；

⑶若後來的不滿意，即再回至原來的樣子。

3. 設計一個FLASHER程式，在第1高解像繪圖頁上畫出一個圖樣，並將黑白反倒的圖樣置於第2高解像繪圖頁內。然後交替輪流顯

示第1頁與第2頁間，使圖樣原形與黑白反倒的圖樣輪流交替顯示。爲了節省時間，程式在連續兩次的POKE顯示開關之間，加上一計數廻路（ FOR I = 1 TO 50 : NEXT I ）否則，顯示幕的速度將無法趕上電腦輸出，以致交替顯示時產生差異。

4. 再設計FLASHER程式，使所有圖樣與反相圖樣（黑白反倒）均發生在顯示文字頁時，只有在執行程式的閃爍時刻，才顯示出繪圖頁。將程式設計成只要收到 ESC 鍵，控制卽脫離閃爍。

5. 設計一個能畫出下列東西的程式：

- 在第2高解像繪圖頁上畫出一間房屋的輪廓（房屋沒有尖角，也沒有門及沒有窗）。
- 房屋圖形不變，但加門加窗，且畫在第3高解像繪圖頁上。
- 將畫在第3頁的圖形，在旁邊加上一個車庫，畫在第4高解像繪圖頁上。
- 將畫在第4頁的圖形，在旁邊加上一棵樹（三角形的聖誕樹），畫在第5高解像繪圖頁上。

全部畫完後，使程式能接收四個按鍵，並在收到每一按鍵後，卽可將其所儲存的圖像呈現。

若您覺得這樣還不夠理想，也可用彩色圖案加上天空，陽光，綠野，車道等等，可能的話也可描出圖樣在春、夏、秋、冬各個季節的景色，去動用您們的頭腦吧！

第三章　繪圖幕文字設計的技巧

APPLESOFT本身並沒有提供在繪圖頁（ graphics pages ）上顯示文字的方法，而只留繪圖頁的底下四行讓使用者來使用，以顯示文字（ text ）。但是此作用並不能夠滿足愛好 APPLE Ⅱ 的使用者，因此本章將介紹二種能在繪圖頁上顯示文字的有效方法，其中例題程式更提供了愛好者在APPLE Ⅱ中多方面的滿足。

點圖型設計的技巧 （ Bit Pattern Text ）

在 Apple Ⅱ參考手册（ Apple Ⅱ Reference Manual）第十四頁裏。明白地告訴我們，在文字頁（ text page ） 裏的每個字元（ character）都是由一“8 × 7”字型的白點所組成，如圖 3.1是字母A的放大字幕。

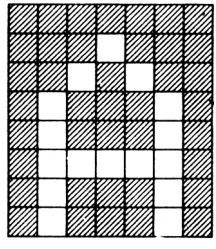


圖 3-1　“A”的放大字幕

而在高解像度（ HGR）繪圖頁裏，它們都是由這些“8 × 7”點型區域（ block ）所組成畫面 280 × 192 個點的繪圖頁，它將顯示幕共分成 24 列（ row ），每列有 40 個這種區域。也因此告訴了我

們，欲在繪圖幕上寫文字，所應注意的範圍。另外文字元的構圖過程則須先提供使用者所希望的標記（ ledel ），而利用點圖型以顯映在繪圖幕上。程式 3.2 的副程式決定了所選擇位元組（ byte ）的記憶位置，須符合所希望的顯示幕位置及顯示點圖型的文字元（ text character ）。

字元群 （Characters）

從第 2 章中知道高解像度是利用存在於某一段電腦記憶體的區域的內容來顯示。

舉例說明，首先考慮記憶位置 8192 ，此RAM的位元組是顯示在高解像度顯示幕時的左上角，假設位元組所存的是資料 67（ POKE 8192,67 ），在二進制法裏 67 表示成（ 01000011 ），而在高解像度幕裏將反次序的存入如：

圖 3-2 內容 “67” 的顯示情形

注意高解像度是使用每位元組中較低的 7 個位元來顯示，而最低的位元（ bit ）顯示在最前面。

同理，假如 8192 所存的是 87（ 二進制 01010111）則一樣地在高解像度顯示幕的左上角顯示如：

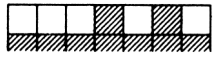


圖 3-3 內容 “87” 的顯示情形

爲讀者說明繪圖幕左上角中 8 個位元組所組成的區域地址如圖 3.4 。

爲了使顯示幕左上角印出數目 “4” ，則我們將安排點圖型的記

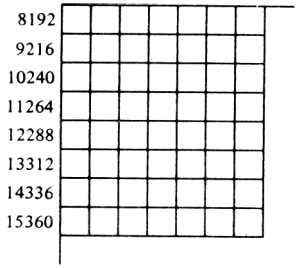


圖 3-4　HGR 幕左上角方塊地址

憶位置如圖 3.5 所示：

	點圖型（反次序） Binary	Decimal
8192	0000000	0
9216	0010000	16
10240	0011000	24
11264	0010100	20
12288	0010010	18
13312	0111110	62
14336	0010000	16
15360	0010000	16

圖 3-5　字幕 “4”

而顯示數目 “4” 的程式命令順序如下：

```
HGR: POKE 8192,0: POKE 9216,16: POKE 10240,24:
POKE 11264,20: POKE 12288,18: POKE 13312,62:
POKE 14336,16: POKE 15360,16
```

我們的工作：首先我們畫點圖型方格表，再將點圖表的二進碼轉換成十進碼，用命令 POKE 到正確的記憶位置。故此，使用者也可以用同樣方法顯示其它數目或符號，而雖然一個簡單的記憶位置只有 “8 × 7” 個點大，但您也可以用數個 “8 × 7” 個點組合顯示較複雜的符號和圖案。

談到這裏，相信讀者會認爲任何希望能顯示的符號，其安排的步驟，是煩瑣及費力的，這是因爲我們在先前設計點圖型程式時缺少一種簡便的技巧，而事實上技巧利用有很多種。程式 3.1 及程式 3.2 即利用陣列 AL％（ I ， J ） 的方法來設計點圖型內容的存取。

由於標準的 Apple Ⅱ字元群有 64 個字元，每個字元顯示都有八個位元組，因此陣列的範圍是 AL％（64，8） 。假設我們數目字元的陣列中第 20 個字元是數目 “4” ，則得到：

AL％（20，0） = 0
AL％（20，1） = 16
AL％（20，2） = 24
AL％（20，3） = 20
AL％（20，4） = 18
AL％（20，5） = 62
AL％（20，6） = 16
AL％（20，7） = 16

因此欲在顯示幕左上角顯示數目 “4” 則須：

```
HGR
FOR I = 0 TO 7
ADDR=8192 + I*1024
POKE ADDR, AL%(20,I)
NEXT I
```

但因 ASCII 碼的最前 32 個值並不符合使用者所欲得的字元，有鑑於此，使用者須先得字元參考變數 K$ 來決定陣列的起點：

```
AL%(CH,0)
```

到

```
AL%(CH,7)
```

這裏

```
CH = ASC(K$) - 32
```

記住：求出要被顯示字元的顯示幕位置。本節先讓我們再複習高

解像度圖案顯示的位址。

看圖 3.4是讓我們了解在顯示高解像度時顯示幕左上角的八位元組區域的位址，它們是從 8192 到 15360 ，而每一位元組，位址就增加 1024 ，到了位址 15360後，則是第二區域的八位元組的開始。而其開始地址是 8320 ，範圍是 8320 到 15488 且每加一位元組，位址也增加 1024 。如此則有八個此類的區域，分佈在顯示幕左邊 1 / 3 處的最上層。而這些區域的其本位址求得：

```
BA = 8192 + 128*I.          ( 0 ≤ I ≤ 7 )
```

而每一區域內的八個位元組 J = 0 ， 1 ， 2 ，…… 7 它們的位址是

```
BA + 1024*J
```

以相似的步驟，可算出顯示幕 1 / 3 處中間層的開始位址，只不過將 8192 替換成 8232（ 注意 8232 = 8192 + 40 ），而 1 / 3 處最下層開始位址則用 8272 替換（ 注意 8272 = 8192 + 2 × 40 ）。

找出了高解像度顯示幕左邊沿次顯示的記憶位置後須再確定：

1. 顯示幕位置是在上，中，或下層間；
2. 每一層間所希望的區域；
3. 每一區域所希望的位元組。

欲得繪圖幕上顯示文字的位址，讀者尤須留意前要點的一、二項，因爲任一數目或符號均是使用區域裏的所有八位元組所組合成，亦即第三項考慮的每一位元組都要用到。

爲了以後讀者的方便，吾將顯示幕的三個層間，分別用層間 0 ，1 ，2 段來表示，而它們同樣的有八個區域於一個層間，而每一區域我們用 0 到 7 來表示（ 由上始到下迄 ）。

程式 3.2 是使用 HT 和 VT （ HTAB 和 VTAB ）來決定欲在顯示幕上文字位置的位址。首先與讀者一起來討論 VT ：

VT = 1～8 是層間 0 的區域

VT = 9～ 16 是層間 1 的區域

VT = 17～24 是層間 2 的區域

由上得知，層間值，SE，的定義：

```
SE = INT((VT - 1)/8).
```

而層間之內的區域（BL = 0～7）亦須由VT 來決定（亦即要決定VT值所代表的是那一個區域），下圖是VT、SE和BL 三者間的關係表：

表 3-1

VT	SE	BL	VT	SE	BL	VT	SE	BL
1	0	0	9	1	0	17	2	0
2	0	1	10	1	1	18	2	1
3	0	2	11	1	2	19	2	2
4	0	3	12	1	3	20	2	3
5	0	4	13	1	4	21	2	4
6	0	5	14	1	5	22	2	5
7	0	6	15	1	6	23	2	6
8	0	7	16	1	7	24	2	7

如果 Applesoft 中有MOD的命令，則區域數 BL =（VT − 1）MOD（8）；若無則

```
BL = VT - 1 - 8*INT((VT-1)/8)
```

談到這裏，我們已可以算出每一區域的其本地址：

```
BA = 8192 + 40*SE + 128*BL
```

因此得知：

```
BA = 8192 + 40*INT((VT-1)/8) +
     128*(VT-1 - INT((VT-1)/8))
```

至目前筆者相信讀者已可很輕易算出在繪圖幕左方沿邊的字元位置的位址：那麼現在我們再一起討論式中加入 HT 的影響。但事實上，位元組的位址與鄰接的水平線僅差 1 而已。由於 VT和 HT 的確知，這位元組的基本位址：

```
BA = 8192 + 40*INT((VT-1)/8) +
     128*(VT-1 - INT(VT-1)/8)) + HT - 1
```

上式了解每個區域基本位置都是由變數 VT和變數 HT 來決定，筆者

現在把上式整理後，我們將又得到如下一個較簡單的相等式：

```
BA = 8063 + 128*VT - 984*INT((VT-1)/8) + HT
```

討　論

當讀者能算出顯示幕上字元的位置之後，現在讓我們再來看點圖型程式的討論與技巧。利用陣列法設計的點圖型是將所需要的符號存置在陣列裏，但定義陣列裏的元素的程式亦是相當累人的一件工作，因爲設計者程式須存入大量的附屬資料（指令）以供程式顯示需要，因此若增大陣列，意謂在文字檔時間上的浪費及磁碟（DISK）記憶空間的徒耗，這是使用者使用陣列點圖型法所應警惕的，下面舉例了一個較簡省時間及記憶空間的方法。

程式3.1是應用了二維整數陣列AL%（I，J）其中I（＝0～63）是以文字變數K$來修正的ASC（K$）－32而決定。組成此文字的八個位元組則均存在於陣列AL%（I，J）內其中J＝0～72，而每一陣列AL%（I，J）僅表示字元內點圖型的一位元組值。因此當陣列讀入資料時，會自動將十進碼轉換成二進碼，而建立了點圖型，始映字元於“8×7”點圖格子上。陣列讀滿後，就會被磁碟保存於二進檔案中（行號90）。

告訴讀者有關陣列保存在磁碟的一些常識：陣列空間的起始位址是107及108（由低位元組開始存）而記憶位址109及110則是陣列空間的結束位址。若有新的變數（如ADDR和LN）時則陣列空間的記憶體須逐次向上移動，因爲簡單的變數是被存在LOMEM和陣列空間之間。正因爲如此，爲了BSAVE時不造成整個陣列的移動，記憶體區域的開始位址（ADDR）及長度（LN）的計算是必須的。而且陣列在被保存之前任何附加的變數，都須先定義，其於此先定義ADDR及LN於行號10。致使行號70及80的計算乃爲了不變亂陣列空間。

程式3.2則說明了陣列的使用，但其本身必須先從磁碟讀取和保

存在適當的記憶位址，並且陣列指標（ array-pointer ）也必須正確的設定。其實陣列的最終目的只是完成ADDR變數的讀取即程式3.2中行號20的結束。緊接陣列讀取（行號30）之後的是BLOAD檔長度的讀取從位址43616（ $AA60 ）及43617（ $AA61 ）。最後告訴讀者一點，行號50及行號60將指標設定在陣列空間結束之時。此時，陣列是隨時被使用者使用了。

程式3.2也表明了符號顯映副程式的使用（行號210～290）。此副程式會接收字串ST$及水平、垂直的標頭（ tab ）設定HT和VT 值，字串是後來一個字元接一個字元讀取。正因如此點圖形圖案也就一個又一個的POKE到我們求出的記憶位置內。

此副程式亦可在高解像顯示幕中應用繪圖程式的標記位置而決定出特定的位置，在跳躍到副程式之前，標記（ ST$）及標頭（ HT 及VT ）的設定是絕對必須的，而設定HT時，要留意所定的標記不能超過右邊的顯示幕，否則因爲將破壞跳行（ line feed），以致標記可能在意料之外的地方出現。同理，肯定HT及VT 的值要在適當的範圍。（ $1 \leq HT \leq 40$ ；$1 \leq VT \leq 24$ ）

程式3·1：點圖字型

```
1   REM PROGRAM 3.1 (BIT PATTERN ALPHABET FILE)
2   REM SAVES AN ALPHABET FILE ON DISK
10   ADDR = 0:LN = 0
20   DIM AL%(64,8)
30   FOR I = 0 TO 63
40   FOR J = 0 TO 7
50   READ AL%(I,J)
60   NEXT J,I
70   ADDR = PEEK (107) + 256 * PEEK (108)
80   LN = PEEK (109) + 256 * PEEK (110) - ADDR
90   PRINT CHR$ (4); "BSAVE ALPHABET, A";
     ADDR;",L";LN
100   END
110   DATA 0,0,0,0,0,0,0,0: REM SPACE
120   DATA 0,8,8,8,8,8,0,8: REM !
130   DATA 0,20,20,20,0,0,0,0: REM "
140   DATA 0,20,20,62,20,62,20,20: REM #
150   DATA 0,8,60,10,28,40,30,8: REM $
```

```
160   DATA 0,6,38,16,8,4,50,48: REM %
170   DATA 0,4,10,10,4,42,18,44: REM &
180   DATA 0,8,8,0,0,0,0,0: REM '
190   DATA 0,8,4,2,2,2,4,8: REM (
200   DATA 0,8,16,32,32,32,16,8: REM )
210   DATA 0,8,42,28,8,28,42,8: REM *
220   DATA 0,0,8,8,62,0,0,0: REM +
230   DATA 0,0,0,0,0,8,8,4: REM ,
240   DATA 0,0,0,0,62,0,0,0: REM -
250   DATA 0,0,0,0,0,0,0,8: REM .
260   DATA 0,0,32,16,8,4,2,0: REM /
270   DATA 0,28,34,50,42,38,34,28: REM 0
280   DATA 0,8,12,8,8,8,8,28: REM 1
290   DATA 0,28,34,32,24,4,2,62: REM 2
300   DATA 0,62,32,16,24,32,34,28: REM 3
310   DATA 0,16,24,20,18,62,16,16: REM 4
320   DATA 0,62,2,30,32,32,34,28: REM 5
330   DATA 0,56,4,2,30,34,34,28: REM 6
340   DATA 0,62,32,16,8,4,4,4: REM 7
350   DATA 0,28,34,34,28,34,34,28: REM 8
360   DATA 0,28,34,34,60,32,16,14: REM 9
370   DATA 0,0,0,8,0,8,0,0: REM :
380   DATA 0,0,0,8,0,8,8,4: REM ;
390   DATA 0,16,8,4,2,4,8,16: REM <
400   DATA 0,0,0,62,0,62,0,0: REM =
410   DATA 0,4,8,16,32,16,8,4: REM >
420   DATA 0,28,34,16,8,8,0,8: REM ?
430   DATA 0,28,34,42,42,26,2,60: REM @
440   DATA 0,8,20,34,34,62,34,34: REM A
450   DATA 0,30,34,34,30,34,34,30: REM B
460   DATA 0,28,34,2,2,2,34,28: REM C
470   DATA 0,30,34,34,34,34,34,30: REM D
480   DATA 0,62,2,2,30,2,2,62: REM E
490   DATA 0,62,2,2,30,2,2,2: REM F
500   DATA 0,60,2,2,2,50,34,60: REM G
510   DATA 0,34,34,34,62,34,34,34: REM H
520   DATA 0,28,8,8,8,8,8,28: REM I
530   DATA 0,32,32,32,32,32,34,28: REM J
540   DATA 0,34,18,10,6,10,18,34: REM K
550   DATA 0,2,2,2,2,2,2,62: REM L
560   DATA 0,34,54,42,34,34,34,34: REM M
570   DATA 0,34,34,38,42,50,34,34: REM N
580   DATA 0,28,34,34,34,34,34,28: REM O
590   DATA 0,30,34,34,30,2,2,2: REM P
600   DATA 0,28,34,34,34,42,18,28: REM Q
```

(continued)

```
610   DATA 0,30,34,34,30,10,18,34: REM R
620   DATA 0,28,34,2,28,32,34,28: REM S
630   DATA 0,62,8,8,8,8,8,8: REM T
640   DATA 0,34,34,34,34,34,34,28: REM U
650   DATA 0,34,34,34,34,34,20,8: REM V
660   DATA 0,34,34,34,42,42,54,34: REM W
670   DATA 0,34,34,20,8,20,34,34: REM X
680   DATA 0,34,34,20,8,8,8,8: REM Y
690   DATA 0,62,32,16,8,4,2,62: REM Z
700   DATA 0,62,6,6,6,6,6,62: REM LEFT BRACKET
710   DATA 0,0,2,4,8,16,32,0: REM BACKSLASH
720   DATA 0,62,48,48,48,48,48,62: REM RIGHT
      BRACKET
730   DATA 0,0,8,20,34,0,0,0: REM ^
740   DATA 0,0,0,0,0,0,0,62: REM UNDERLINE
```

程式 3·2：繪圖頁的點圖型文字

```
1   REM PROGRAM 3.2 (BIT PATTERN TEXT FOR THE
    GRAPHICS PAGE)
2   REM DEMONSTRATES PRINTING FOR THE GRAPHICS
    PAGE
10  ADDR = 0;LN = 0
20  ADDR = PEEK (107) + 256 * PEEK (108)
30   PRINT CHR$ (4);"BLOAD ALPHABET,A";ADDR
40  LN=256 * PEEK (43617) + PEEK (43616)
50   POKE 110, INT ((ADDR + LN) / 256)
60   POKE 109,ADDR + LN - 256 * INT ((ADDR + LN)
     / 256)
70   HGR : HOME : VTAB 21
80  VT = 5:HT = 5:ST$ = "BIT PATTERN TEXT FOR
    HGR"
90   GOSUB 210
100 VT = 7:HT = 5:ST$ = "PROVIDING TEXT,
    NUMBERS, AND SYMBOLS"
110  GOSUB 210
120 VT = 10:HT = 10:ST$ = "TRY IT!"
130  GOSUB 210
140 VT = 20:HT = 1:ST$ = "ENTER:"
150  GOSUB 210
160  INPUT "STRING ";ST$
170  INPUT "VTAB ";VT
180  INPUT "HTAB ";HT
190  GOSUB 210
200  END
210 BA = 8063 + 128 * VT - 984 *
    INT ((VT - 1) / 8) + HT
220  FOR I = 1 TO LEN (ST$)
```

```
230 CH = ASC ( MID$ (ST$,I,1)) - 32
240  FOR J = 0 TO 7
250  POKE BA + 1024 * J,AL%(CH,J)
260  NEXT J
270 BA = BA + 1
280  NEXT I
290  RETURN
```

注意與建議

1. 我們可利用程式而使用點圖型的方法先爲繪圖頁預備一些符號或圖案，使用者也可以經過思考設計一些典型的字元及以後可能增加的符號而幫助它功能的發揮。另加說明的設計的程式若用機器語言來執行的話，作圖速度也將較過去來得快喔！市面上亦有人在賣以“點圖型”方式在繪圖顯示幕上顯示符號或圖案的程式，且是用機器語言形式來設計，可供讀者來選購。

2. 讀者若欲將程式3.2的副程式中的顯映標記由水平修改成垂直，則可以如下

```
210   FOR I = 1 TO LEN (ST$)
220   BA = 8063 + 128 * VT - 984 *
      INT ((VT - 1) / 8) + HT
230 CH = ASC ( MID$ (ST$,I,1))-32
240   FOR J = 0 TO 7
250   POKE BA + 1024 * J,AL%(CH,J)
260   NEXT J
270 VT = VT + 1
280   NEXT I
290   RETURN
```

若許你覺得將此程式 3.2改成具有 2 個副程式，一則水平標記一則垂直標記會比較好這也是可以的。

3. 讀者若欲獲得一反向（ inverse ）顯示可將程式 3.2 中行號 250 改成：

```
250 POKE BA + 1024*J, 255 - AL%(CH,J)
```

4. 如果讀者希望顯示倒置的符號可將程式 3.2 中行號 250 改成：

```
250 POKE BA + 1024*J, AL%(CH,7 - J)
```

5. 欲充實程式 3.2副程式的跳行（回列）的能力，須改變行號 210～290 如下：

```
210  FOR I = 1 TO LEN (ST$)
220 BA = 8063 + 128 * VT - 984 *
    INT ((VT - 1) / 8) + HT
230 CH = ASC ( MID$ (ST$,I,1)) - 32
240  FOR J = 0 TO 7
250  POKE BA + 1024 * J,AL%(CH,J)
260  NEXT J
270 HT + HT + 1: IF HT = 41 THEN HT = 1:
    VT = VT + 1
280  NEXT I
290  RETURN
```

而要保護一個不希望是在中間空斷的字或敍述，可試著增加一廻路（ loop ）（於字元與空間之間），並要計算其長度，但如果此字的長度超過 40-HT，則它會超過現行（ current line）則必須再加上回列（ line feed ）如下：

```
HT = 1:VT = VT + 1: GOTO ***
```

6. 讀者也可以將陣列 AL%（ I ， J ）擴大，來增加一些小寫的英文字母及特殊符號。如果陣列的範圍是 AL%（ 90,8），則我們可將那些特殊字母的點圖型設定在起點 AL%（ 64,J ）到 AL%（ 89, J ）處。另若想利用現存的副程式來顯示標記 “Lower ” 的字串時可：

```
ST$ = "L" + CHR$(78) + CHR$(86) +
CHR$(68) + CHR$(81)
```

7. 欲定義的字串內容有數目時，如：

```
ST$ = "X = 45"
```

好好的利用這個副程式，欲顯示程式裏面被計算的數目 X 時，請使用

```
ST$ = STR$(X)
```

8. 利用第 2 高解像繪圖頁（ HGR2 ）顯示時亦能很有效地來完成，但其 RAM 位置的地址須符合顯示幕左上角的位址改變成 8192 到 16384 因此行號 210 須改成：

```
210 BA = 16255 + 128 * VT - 984 *
INT ((VT - 1)/8) + HT
```

9. 此程式雖不允許直接從鍵盤輸入某些符號（如“，”）不過若這些符號是包含在字串內的話如：

```
ST$ = "A,B"
```

則此副程式將會很適當地處理這些符號的，請讀者放心。

造型表設計的技巧（Shape Table Text）

現在爲讀者再介紹另一種在繪圖頁上顯示文字的方法，我們叫做“造型表”法。筆者以爲造型表比先前使用的點圖型更來得實用及適合讀者。但它也較點圖型法須付出更多的事前準備了。而以下就與讀者一同討論“造型表”文字的使用設計技巧。

造型表是事先存集好形狀或繪圖案，並給與它們指定的位置。需要它們時，就能夠很迅速地顯示在高解像度繪圖頁上。而它們的圖案和選擇均是用數碼來記存。至於指定顯示的圖樣號碼N和其顯示的位置（X，Y）常用到的指令是：

```
DRAW N AT X,Y
```

或

```
XDRAW N AT X,Y
```

因此利用DRAW或XDRAW可在希望的位置上顯示指定的圖案。而所不同的是DRAW是顯示最近HCOLOR命令的顏色，而XDRAW則是其顏色的補色，這也就是二個連續的XDRAW命令可造成圖案擦掉的理由

```
XDRAW N AT X,Y        （補色）
XDRAW N AT X,Y        （再一次補色）
```

二次補色後能擦掉圖案，而且不會影響背景。

在Applesoft參考手册裏對造型表的設計及其構成有概略地說明。而雖然使用者設計及圖案程式的處理過程是冗長及枯索的工作，但

事實上，此工作由電腦來做是最適合了。在許多的電腦雜誌、亦曾經介紹過造型表的程式。在附錄 2 中有詳細的舉例造型表的安排與設計，並對圖形作了一番概略的討論。

在本節，我們則對造型表文字的顯示感到興趣。首先造型表要有標準文字符號的圖案，這也是建立造型表的第一步。程式 3.3的造型表包含了標準的文字頁字元，而其中的數目圖型是用ASCII 字碼如：

圖型字碼K$ = ASC(K$) − 32

程式 3·3：造型表字型檔

```
1   REM PROGRAM 3.3 (SHAPE TABLE ALPHABET FILE)
2   REM SAVES A SHAPE TABLE ALPHABET ON DISK
10   FOR I = 24576 TO 25364
20   READ X
30   POKE I,X
40   NEXT I
50   PRINT CHR$ (4);"BSAVE SHAPE ALPHABET,
     A24576,L793"
60   END
70   DATA 63,0,128,0,134,0,142,0,157,0:
     REM                                  ----------
80   DATA 172,0,185,0,200,0,206,0,214,0
90   DATA 222,0,236,0,245,0,249,0,255,0
100   DATA 2,1,9,1,24,1,32,1,44,1:
      REM                                 (造型表)
110   DATA 56,1,67,1,79,1,93,1,102,1
120   DATA 116,1,127,1,133,1,139,1,148,1
130   DATA 156,1,165,1,175,1,190,1,202,1:
      REM                                 (定址)
140   DATA 216,1,228,1,240,1,254,1,8,2
150   DATA 19,2,31,2,40,2,48,2,61,2
160   DATA 70,2,82,2,94,2,106,2,116,2:
      REM                                 (結構)
170   DATA 130,2,143,2,155,2,163,2,174,2
180   DATA 185,2,198,2,210,2,219,2,231,2
190   DATA 245,2,253,2,9,3,17,3:
      REM                                 ----------
200   DATA 73,4,32,36,36,0: REM !
210   DATA 9,64,24,32,108,54,4,0: REM "
220   DATA 9,36,103,60,5,32,13,246,45,23,30,45,
      23,38,0: REM #
```

```
230  DATA 1,40,53,12,12,28,55,28,28,12,37,22,12,
     37,0: REM $
240  DATA 73,41,60,223,12,12,12,5,248,35,55,4,0:
     REM %
250  DATA 73,9,28,28,28,28,100,21,190,26,174,
     101,8,4,0: REM &
260  DATA 73,64,24,32,36,0: REM '
270  DATA 73,28,28,36,12,12,4,0: REM (
280  DATA 73,12,12,36,28,28,4,0: REM )
290  DATA 73,36,36,36,141,23,31,28,150,98,13,21,
     4,0: REM *
300  DATA 73,32,60,111,41,31,32,4,0: REM +
310  DATA 9,12,36,0: REM ,
320  DATA 64,24,41,45,37,0: REM -
330  DATA 73,4,0: REM .
340  DATA 1,96,12,12,12,4,0: REM /
350  DATA 9,45,12,36,36,28,63,23,54,54,12,12,12,
     4,0: REM 0
360  DATA 9,45,28,36,36,188,4,0: REM 1
370  DATA 73,9,63,63,100,12,101,228,63,23,4,0:
     REM 2
380  DATA 8,21,45,12,228,103,5,32,63,63,4,0:
     REM 3
390  DATA 73,33,44,31,63,100,12,12,54,38,0:
     REM 4
400  DATA 8,21,45,12,36,28,63,39,44,45,37,0:
     REM 5
410  DATA 9,45,12,228,63,55,38,64,3,12,12,45,
     4,0: REM 6
420  DATA 9,36,12,12,12,60,63,39,0: REM 7
430  DATA 9,45,12,228,63,23,38,64,3,100,45,21,
     38,0: REM 8
440  DATA 41,101,12,36,228,63,23,174,45,4,0:
     REM 9
450  DATA 73,64,3,4,32,0: REM :
460  DATA 9,5,32,4,32,0: REM ;
470  DATA 73,225,28,28,12,12,12,4,0: REM <
480  DATA 64,45,45,4,56,63,39,0: REM =
490  DATA 9,12,12,12,28,28,28,4,0: REM >
500  DATA 73,4,32,12,12,28,63,23,4,0: REM ?
510  DATA 9,45,37,64,3,36,28,63,23,54,54,76,229,
     36,0: REM @
520  DATA 33,36,100,12,14,14,54,63,111,17,38,0:
     REM A
530  DATA 33,36,36,44,45,21,190,31,109,50,23,63,
     4,0: REM B
540 DATA 73,9,184,63,28,36,36,12,45,21,4,0:
    REM C
550 DATA 33,36,36,44,45,21,54,54,23,63,4,0:
    REM D
```

(continued)

```
560 DATA 41,45,37,192,63,55,38,64,3,36,45,
    45,4,0: REM E
570 DATA 33,36,36,44,45,181,26,63,4,0: REM F
580 DATA 9,45,37,60,223,34,36,100,45,37,0:
    REM G
590 DATA 33,36,36,180,10,45,37,36,150,50,38,0:
    REM H
600 DATA 9,45,28,36,36,60,13,4,0: REM I
610 DATA 1,168,45,12,36,36,36,0: REM J
620 DATA 33,36,36,108,9,23,23,23,21,21,21,4,0:
    REM K
630 DATA 73,9,63,63,36,36,36,4,0: REM L
640 DATA 33,36,36,172,21,102,96,54,54,54,4,0:
    REM M
650 DATA 33,36,36,172,170,21,149,36,36,36,4,0:
    REM N
660 DATA 9,45,12,36,36,28,63,23.54,54,4,0:
    REM O
670 DATA 33,36,36,44,45,21,190,63,4,0: REM P
680 DATA 9,37,168,21,4,32,36,28,63,23,54,
    54,4,0: REM Q
690 DATA 33,36,36,44,45,21,190,63,21,21,21,4,0:
    REM R
700 DATA 8,21,45,12,60,56,231,100,45,21,4,0: REM
    S
710 DATA 73,36,36,36,63,77,37,0: REM T
720 DATA 9,45,12,36,36,252,27,54,54,38,0:
    REM U
730 DATA 73,12,12,36,36,223,51,54,174,4,0:
    REM V
740 DATA 33,36,36,108,9,54,54,54,28,28,
    180,35,0: REM W
750 DATA 33,12,12,28,28,108,9,190,22,21,38,0:
    REM X
760 DATA 73,36,228,28,108,9,246,4,0: REM Y
770 DATA 73,9,63,63,100,12,12,12,60,63,39,0: REM
    Z
780 DATA 73,9,63,63,44,60,44,60,44,28,45,
    45,4,0: REM LEFT BRACKET
790 DATA 72,73,28,28,28,28,4,0: REM BACKSLASH
800 DATA 41,45,37,39,37,39,37,39,253,63,4,0: REM
    RIGHT BRACKET
810 DATA 64,24,97,12,21,21,4,0: REM ^
820 DATA 41,45,37,0: REM UNDERLINE
```

對使用造型表顯示文字讀者應該注意如：

1 造型表元素的可用率（使用比）；

2 造型表文字顯示的鑑別；

3. 顯示圖案所在位置的鑑別。

程式3.4將舉例說明：

程式3·4：繪圖頁的造型表文字

```
1  REM PROGRAM 3.4 (SHAPE TABLE TEXT FOR THE
   GRAPHICS PAGE)
2  REM DEMONSTRATES HIGH RES PRINTING WITH
   SHAPES
10  PRINT CHR$ (4);"BLOAD SHAPE ALPHABET,
    A24576"
20  POKE 232, PEEK (43634): POKE 233,
    PEEK (43635)
30  HOME : HGR : HCOLOR= 3: VTAB 21: SCALE= 1:
    ROT= 0
40 VT = 5:HT = 5:ST$ = "SHAPE TABLE TEXT FOR
   HGR"
60  GOSUB 180
70 VT = 7:HT = 5:ST$ = "PROVIDING TEXT, NUMBERS,
   AND SYMBOLS"
80  GOSUB 180
90 VT = 10:HT = 10:ST$ = "TRY IT!"
100  GOSUB 180
110 VT = 20:HT = 1:ST$ = "ENTER:"
120  GOSUB 180
130  INPUT "STRING ";ST$
140  INPUT "VTAB ";VT
150  INPUT "HTAB ";HT
160  GOSUB 180
170  END
180 HT = 7 * (HT - 1):VT = 8 * VT - 1
190  FOR I = 1 TO LEN (ST$)
200 CH = ASC ( MID$ (ST$,I,1)) - 32
210  IF CH = 0 THEN 230
220  XDRAW CH AT HT,VT
230 HT = HT + 7
240  NEXT I
250  RETURN
```

行號10 是從磁碟讀取得來的造型表，其造型表的記憶位置是任意定址的（但千萬不能存入程式上方或任一繪圖頁中）。行號20 則指明了造型表的位置，由此才可以存取XDRAW 的命令，另最新的BLOAD檔案的目的位置的地址是43634和43635 。

通常，造型表位置的低位元組地址和高位元組地址是232和233

。舉例欲使位址A之造型表可以存入之敍述如下：

```
POKE 232, A-256*INT(A/256): POKE 233, INT(A/256)
```

程式中行號30置定SCALE = 1和ROT = 0 它們控制了圖形大小及旋轉位置。在Applesoft 程式參考手册中有詳細討論。但程式3.4則用HT及VT 的值來定作圖座標。以上就是利用造型表在繪圖頁上顯示文字的程式步驟，於此筆者僅作大致的介紹，但希望讀者能參閱Apple Ⅱ 參考手册及附錄2，以達舉一反三之效是筆者與讀者之望也。

注意與建議

1. 程式3.4中的圖案是用HT及VT定位置，就好像文字頁中用HTAB及VTAB的功能一樣。文字只允許有24列，每列40 個字元，而假如消除行號180的敍述，則VT及HT之值，可定 $0 \leq VT \leq 191$ 及 $0 \leq HT \leq 279$。也就是允許更多合適的位置標誌。

程式3.3中，其字元的實際位置（ HT ， VT ），是左方底邊一個“8 × 7”點的字型。注意XDRAW命令將“紮圍”（ wrap around ）顯示幕的圖案它是：

```
XDRAW 33 AT 276,10
```

（ 276,10 ）象徵了左下角一個“8 × 7”點的字型，當字型超過了顯示幕右方邊界時，將被顯示幕“紮圍”而呈現在左方。同理，圖樣過近於顯示幕頂端時也會被“紮圍”到顯示幕的底端了。

2. 造型表中造型圖也可以比例調整大小。程式3.4裏行號40 SCALE = 1替換成SCALE = 2或SCALE = 3時的情形變化。行號230是調整SCALE大於1時須重新調整字元的位置，當SCALE 增大時，則繪圖向量會繪出更多的點，圖形也就愈鬆弛。

3. 造型表中造型圖也可以旋轉，程式3.4中行號40 ROT = 0如改替換成ROT = 16 ，ROT = 32或ROT = 48時則造型圖將依順時鐘方向旋轉90°，180°，或270° 。

4. 欲調整副程式中的垂直標記，只須改行號 230 ：

```
230 VT = VT + 8
```

讀者可有辦法使字元顯示成對角線方向？或顛倒、向後顯映呢？

5. 造型表文字與前面的點圖型文字作比較各有它們的優點與否。而造型表優點之一即是很容易地在不影響繪圖幕顯示的其它部份下擦掉標記，而標記的更換可能是動態圖案的標記或任何其它符號的標記。而動態造形第 8 章有例題。屆時再與讀者詳談。

6. 至少有關點圖型的聯想，如數目的點圖型顯示，或小寫字母的字元點圖型，均可通用於造型文字表中。

第三部份

平面圖形

第四章　電腦繪圖的軟性技巧

本章的目的是告訴讀者有關微電腦繪圖的程式設計者與微電腦本身所提供的道具（ tools ）之間所進行的一項交易行爲及運用技巧。每項的道具將以解釋“如何”繪圖及“爲什麼”的方式，針對讀者所感興趣的而加以介紹。例題程式的舉例說明也暗示我們使用它們的技巧及技術上的探討。而本章所例舉的道具，均是筆者於實際經驗上曾接觸到的。現與讀者共同來學習、分析及討論。而最終的盼望是讀者的認同與喜愛。

顯示幕比例（ Scaling The Screen ）

相信我們已能夠共同且清楚的發現水平和垂直方向的比例在微電腦顯示幕中兩者的差異，而在 Apple Ⅱ 裏它們建立了 280 × 192 的小方格，我們暫時用單位的觀念來比喻。一個水平單位相當於 192 / 280個垂直單位。因此讀者要有一項認同由於比例刻度劃分的不同對於顯示幕的調整也不同，影響也較大。

至於正方型有相等的邊；圓周上的各點至中心點等距的情況來說，我們就希望水平與垂直方向的比例能相同。因此繪圖顯示幕的軟性道具中水平與垂直比例的考慮是重要的。因此決定繪圖之前，比例值的要求是須定義的。

因爲顯示裝置的調整，會影響比例大小的改變，因此我們有必要測定電腦組合顯示幕（即電腦與顯示幕間的連線組合）的比例因素（ scaling factor ）。下面舉出方法來說明比例因素，如下：

1 畫一正方型。

2. 量“正方型”的高（H）和寬（W）。

3. 比例因素（SC）= H / W。

4. 畫圖之前，水平座標乘 SC。

實例：

1.

```
10 HGR2: HCOLOR = 3
20 HPLOT 20,20 TO 180,20 TO 180,180 TO 20,180
TO 20,20
```

2. “正方形”在顯示幕測量結果：寬是 14.1cm 高是 16.3cm。

3. SC = H / W = 16.3 / 14.1 ≈ 1.16

4. 較正的“正方形”

```
10 HGR2: HCOLOR = 3: SC = 1.16
20 HPLOT 20*SC,20 TO 180*SC,20 TO 180*SC,180
   TO 20*SC,180 TO 20*SC,20
30 END
```

請讀者注意 SC 值視使用電腦顯示器的組合情況而定。

在許多應用中，我們可以不必注意水平和垂直刻度的是否相等。通常，在這書裏程式均未提供相等的刻度，但讀者若希望有相等的比例亦是可以很輕易獲得。

畫面的截割及觀測台 （Clipping And Viewports）

當我們指定要畫的點（points）或線（lines）時，它就得在顯示幕可見的地方，很小心映所有的點或線段。我們暫用 Apple BASIC 命令執行：

```
HPLOT 90,25 TO 360,175
```

但顯示幕回答訊息：

```
?ILLEGAL QUANTITY ERROR
```

那是因爲（360，175）這點已經座落於超過顯示幕的右緣。因此爲了線段能顯示在欲見的部份時我們必須自己計算或由電腦計算出線的端點與顯示幕緣的交點座標。

假設線的等式是

$$Y = 25 + (X - 90)(175 - 25)/(360 - 90), \text{ or } Y = -25 + 5(X/9)$$

因爲交點定發生在右界X = 279處，因此Y = − 25 + 5（ 279 / 9 ）所以我們可求得線段與顯示幕界的交點是在（ 279，130 ），因此使用命令

```
HPLOT 90,25 TO 279,130
```

截　割（Clipping）

截割指示是將超過顯示幕邊界的點及線段截割掉而捨棄，而只顯示處理可見部份的點及線段。將獨立的點截割是相當容易——點座標在顯示幕邊界內的則“畫”；座落在邊界之外時則“不畫”如此而已。

在Applesoft BASIC中乃利用如下：

```
100 IF X < 0 OR X > 279 OR Y < 0 OR Y > 191
    THEN 120
110 HPLOT X,Y
120
```

而“線段”的截割是比“點”的截割稍複雜一點，假如線段的某一端點或二個端點座落在顯示幕之外，則我們必須確定顯示幕界與線段的交接點，利用此交點而畫線段可見的部份。

顯示幕被邊界的四條線分割成九塊區域（圖4.1），首先我們定義：線段的兩個端點必於座落在這九塊區域。如果二個端點落在中央區域，則此線段可被顯示，且可以畫出，否則其它區域時，則必須重新測定線段可顯示的部份。

線段與顯示幕邊界的交接點必須要由一個直線截割的副程式來檢視，交接點確立之後，座落在顯示幕內的線段則畫上，其餘部份捨棄。

應用程式4.1所提供的截割副程式來劃線段以圍一個矩形。由分

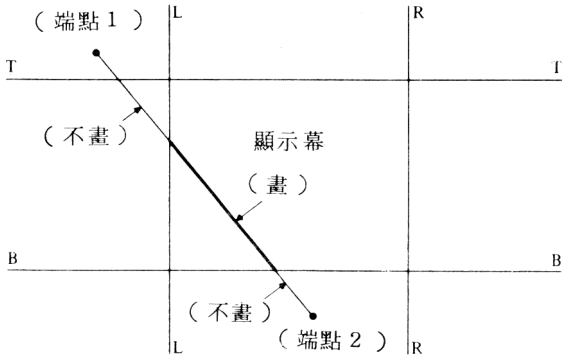


圖 4-1 顯示幕平面區域

類的DATA 敍述來定其端點（行號 300 至 330 ）。因爲端點 X(1) Y(1) 由四個變數來決定其狀況，所以假如點X(1)，點Y(1) 座落在L，R，T和B的顯示幕內側時，則 L(1)，R(1)，T(1)，B(1) 的陣列值被設定爲“0”；反之其值被設定爲“1”。同理可設定L(2)，R(2)，T(2)，B(2) 的陣列值。至於端點情形的陣列值由副程式行號 100 到 150 來定。

舉例，圖4.1 的點位置來定陣列值如下

T(1) = 1　L(1) = 1　B(1) = 0　R(1) = 0

T(2) = 0　L(2) = 0　B(2) = 1　R(2) = 0

行號 200 − 250是檢查線段是否該捨棄的副程式。假如二個端點都在邊界外側相當於L(1) × L(2)，R(1) × R(2)，T(1) × T(2) 或B(1) × B(2) 爲“1”，否則爲“0”，因此如果 L(1) × L(2) + R(1) × R(2) + T(1) × T(2) + B(1) × B(2) 之和不爲“0”則此線段可被捨去（行號 200 ）。

行號 210是如果端點座落在所有邊界之內側時則可劃出線段。否則的話表示定有一端點座落在其中一條邊界線的外側。

行號 220 到 250 是執行使線段與某一顯示幕界的交接點上置點

X(I)，Y(I)。而此被置上的新端點會再回到CLASSIFY 副程式重新執行判斷，後再回到CLIP副程式而截割。

程式對線段的截割會不停的檢視直到行號210將二個端點顯示（線段被劃）或直到行號200二端點座落在顯示幕之外（線段被捨棄）。

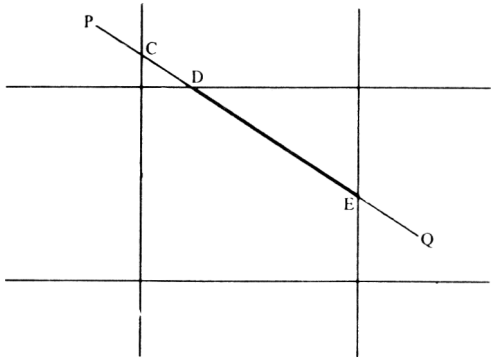


圖4-2　截割說明

圖4.2線段PQ用程式4.1執行後。首先截割成CQ再成DQ，最後截割成DE，而將劃線段DE。

程式4.1我們就是利用此種將線段慢慢截割的演算法來作圖。而如果線段而複雜，而此種演算會很慢。所以如果讀者假設所有的線段都落在顯示幕內，那我們也就不須利用此種演算法來分析了。如果一定要演算分析的話，也可利用程式5.7的方法。雖然沒有程式4.1來得完善，但速度較快，且程式也較簡化。

程式4·1：截割

```
1  REM PROGRAM 4.1 (CLIPPING)
2  REM EXAMPLE OF USE OF CLIPPING SUBROUTINE
10 XR = 279:XL = 0:YT = 0:YB = 191
```

```
20  HGR2 : HCOLOR= 3:SC = 1.16
25  REM ****MAIN LOOP
30  FOR J = 1 TO 4: FOR I = 1 TO 2
40  READ X(I),Y(I): GOSUB 100
50  NEXT I
60  GOSUB 200
70  NEXT J
80  END
95  REM **** CLASSIFY SUBROUTINE
100 L(I) = 0:R(I) = 0:T(I) = 0:B(I) = 0
110  IF X(I) < XL THEN L(I) = 1
120  IF X(I) > XR THEN R(I) = 1
130  IF Y(I) < YT THEN T(I) = 1
140  IF Y(I) > YB THEN B(I) = 1
150  RETURN
190  GOSUB 100
195  REM **** CLIPPING SUBROUTINE
200  IF L(1) * L(2) + R(1) * R(2) + T(1) * T(2)
     + B(1) * B(2) <  > 0 THEN RETURN
210 I = 1: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN I = 2: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN HPLOT X(1),Y(1) TO X(2),Y(2): RETURN
220  IF L(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XL - X(1)) / (X(2) - X(1)):
     X(I) = XL: GOTO 190
230  IF R(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XR - X(1)) / (X(2) - X(1)):
     X(I) = XR: GOTO 190
240  IF T(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YT - Y(1)) / (Y(2) - Y(1)):
     Y(I) = YT: GOTO 190
250  IF B(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YB - Y(1)) / (Y(2) - Y(1)):
     Y(I) = YB: GOTO 190
300  DATA 20,-20,300,40
310  DATA 20,-20,-10,120
320  DATA -10,120,270,180
330  DATA 300,40,270,180
```

觀測台 （Viewports）

顯示幕就如同一個讓我們看到圖像的觀測台（ viewport ）。也就是截割掉觀測台上看不見的部份。而我們對顯示幕的顯示感興趣的部份，也就是觀測那觀測台上可見的部份。

設定變數 XR ， XL ， YT 和 YB 的方法，我們即可把截割法的副程式改作觀測台副程式。例如， XR = 120 ， XL = 25 ， YT = 130

和YB = 180 則此副程式如同觀測台將在顯示幕較左下角的地方顯示可見部份的圖像。

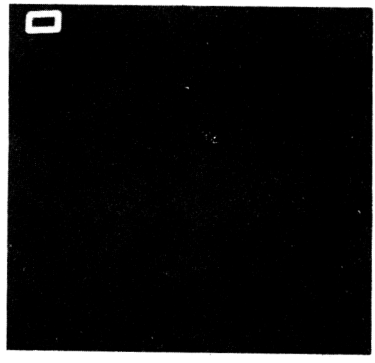
變　換（Transformations）

簡　介

程式 4.2是利用Apple 顯示幕的左上角部份繪一個 20 × 10 的矩形。

程式 4·2：矩形之一

```
1   REM PROGRAM 4.2 (RECTANGLE 1).
2   REM PLOTS ONE RECTANGLE
10   HOME : HGR : HCOLOR= 3
20   HPLOT 0,0 TO 20,0 TO 20,10 TO 0,10 TO 0,0
```



或程式 4·3：矩形之二

```
1   REM PROGRAM 4.3 (RECTANGLE 2)
2   REM PLOTS ONE RECTANGLE
10   HOME : HGR : HCOLOR= 3
20   READ X,Y: HPLOT X,Y
30   FOR I = 1 TO 4
40   READ X,Y: HPLOT TO X,Y
```

```
50   NEXT I
60   DATA 0,0,20,0,20,10,0,10,0,0
```

雖然程式4.3較長但却是二者之中較能爲讀者學一反三的程式。由DATA敍述的變更而順序連接每一線段，當然，4.2的程式也具有相同的能力。但我們使用程式4.3是以較遠的利益的眼光來看且樂易接受的。

程式4.4完成了8個矩形的圖像，而每一像的完成都是用命令DATA來執行。在程式4.3中是用X，Y的值來繪點或線段(HPLOT TO X,Y)。程式4.4則利用座標(X，Y)的變換而求得顯示幕座標(SX,SY)來繪圖。至於長方形的大小、圖樣及位置均由A，B，C和D的值變換來控制。

程式4.4說明了兩個代表型的變換法。RECTANGLES：

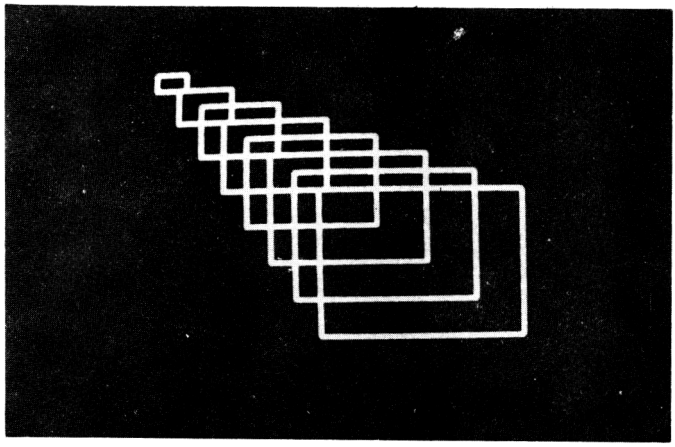
1. 座標的長度乃由A和C的乘積來決定。說明圖像的比例。
2. 圖像的位置乃由B和D的值來決定。說明圖像的平移。

爲了使讀者更熟悉比例法和平移法變換，試下面：

1. 寫一個大小不同的八個矩形，而都有一個頂點顯示在顯示幕左上角(用比例法，不用平移法)。

程式4·4：矩形變化

```
1   REM PROGRAM 4.4 (RECTANGLES)
2   REM PLOTS 8 RECTANGLES FROM 1 SET OF DATA
10 A = 20:B = 0:C = 10:D = 0
20   HOME : HGR : HCOLOR= 3
30   FOR J = 1 TO 8
40   READ X,Y:SX = A * X + B:SY = C * Y + D
50   HPLOT SX,SY
60   FOR I = 1 TO 4
70   READ X,Y:SX = A * X + B:SY = C * Y + D
80   HPLOT TO SX,SY
90   NEXT I
100 A = A + 15:B = B + 15:C = C + 10:D = D + 10
110   RESTORE
120   NEXT J
130   DATA 0,0,1,0,1,1,0,1,0,0
```



2. 寫一個大小相同的八個矩形，而都顯示在顯示幕不同的地方（用平移法，不用比例法）。

變換的便利，通常在繪圖時才看得出。而談到變換筆者想該介紹一些有關連的話題：典型的點“矩陣”和“均質坐標”的表示法。

何謂矩陣（matrix）？

矩陣是矩形的數目陣列，如下：

$$A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\quad 或\quad B=\begin{pmatrix}0 & 2\\ 5 & -3\end{pmatrix}$$

矩陣可相加

$$A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}+\begin{pmatrix}0 & 2\\ 5 & -3\end{pmatrix}=\begin{pmatrix}1+0 & 2+2\\ 3+5 & 4-3\end{pmatrix}=\begin{pmatrix}1 & 4\\ 8 & 1\end{pmatrix}$$

矩陣可相乘

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\begin{pmatrix}0 & 2\\ 5 & -3\end{pmatrix}=\begin{pmatrix}(1)(0)+(2)(5) & (1)(2)+(2)(-3)\\ (3)(0)+(4)(5) & (3)(2)+(4)(-3)\end{pmatrix}$$

$$= \begin{pmatrix} 10 & -4 \\ 20 & -6 \end{pmatrix}$$

至於有關矩陣的符號說明及計算請見附錄 4 有進一步的介紹。

變換的矩陣型態 （ Matrix Representation Of Transformations ）

"點" 在二維平面的變換可用矩陣的型態：點（ 1 × 2 ）矩陣與 2 × 2 矩陣的乘積來表示。例如我們可以變換（移動）點（ X ，Y ）到新點（ X′ ，Y′ ）如下：

$$X \ , \ Y \begin{pmatrix} A & B \\ C & D \end{pmatrix} = (\ AX + CY \ , \ BX + DY \ ) = (\ X' \ , \ Y' \ )$$

因此，向讀者定義新位置——變換點的表示

$(\ X \ , \ Y \ )' = (\ X' \ , \ Y' \ ) = (\ AX + CY \ , \ BX + DY \ )$

由上很明顯地曉得，變換結果是由所選擇的A ，B ，C ，和D的值來測定。程式 4.5容納了A ，B ，C和D的值，而且是以正方形的變換點來作其說明。所以筆者欲以此程式舉例說明其它數種的變換。

本體法（ identity ）

矩陣型式

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

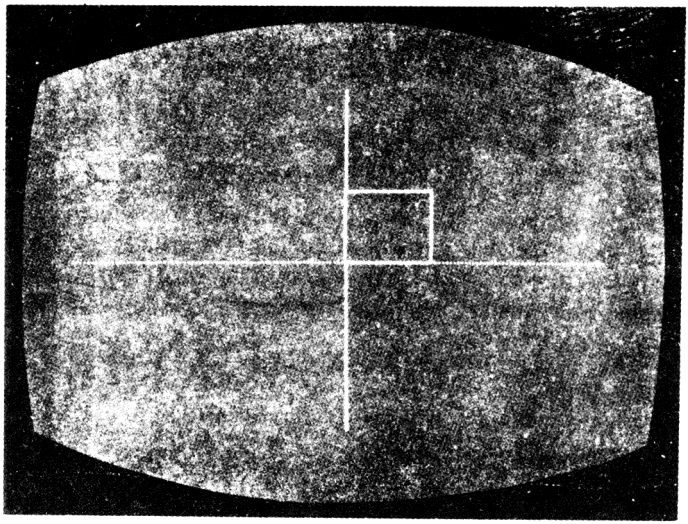
表示了本體變換法（ 也可叫做相等變法 ）

因爲

$$(\ X \ , \ Y \ ) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (\ X \ , \ Y \ )$$

因此本體變換來改變其置點（ leaves points ）。

執行程式4.5和令A＝1，B＝0，C＝0和D＝1後。將得到一個本體變換法改變的正方形，



程式4.5本體變換

程式4·5：矩陣

```
1   REM PROGRAM 4.5 (MATRIX)
2   REM SHOWS EFFECT OF A 2x2 TRANSFORMATION
    MATRIX
10  CX = 140:CY = 96:SC = 1.16
20  XR = 279:XL = 0:YT = 0:YB = 191
30   TEXT : HOME
40   INPUT " A = ";A
50   INPUT " B = ";B
60   INPUT " C = ";C
70   INPUT " D = ";D
80   HGR : HCOLOR= 6
90   HPLOT 0,96 TO 279,96: HPLOT 140,0 TO 140,
     191
100   HCOLOR= 3
110   FOR S = 1 TO 4
120   FOR I = 1 TO 2
130   READ X,Y
```

```
140 TX = A * X + C * Y:TY = B * X + D * Y
150 X(I) = SC * TX + CX:Y(I) = CY - TY
160  GOSUB 300
170  NEXT I
180  GOSUB 380
190  NEXT S: RESTORE
200  VTAB 21: PRINT "A B"; TAB( 11);A;
     TAB(20);B
210  VTAB 22: PRINT TAB( 8);"="
220  VTAB 23: PRINT "C D"; TAB( 11);C;
     TAB(20);D
230  VTAB 24: PRINT "PRESS 'N' FOR NEW MATRIX;
     'ESC' TO END";
240  VTAB 10: GET A$: IF A$ = "N" THEN 30
250  IF ASC (A$) = 27 THEN TEXT : HOME : END
260  GOTO 240
270  DATA 0,0,40,0,40,0,40,40,40,40,0,40,0,40,
     0,0
280  END
290  REM **** CLASSIFY SUBROUTINE
300 L(I) = 0:R(I) = 0:T(I) = 0:B(I) = 0
310  IF X(I) < XL THEN L(I) = 1
320  IF X(I) > XR THEN R(I) = 1
330  IF Y(I) < YT THEN T(I) = 1
340  IF Y(I) > YB THEN B(I) = 1
350  RETURN
360  GOSUB 300
370  REM **** CLIPPING SUBROUTINE
380  IF L(1) * L(2) + R(1) * R(2) + T(1) * T(2)
     + B(1) * B(2) < > 0 THEN RETURN
390 I = 1: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN I = 2: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN HPLOT X(1),Y(1) TO X(2),Y(2): RETURN
400  IF L(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XL - X(1)) / (X(2) - X(1)):
     X(I) = XL: GOTO 360
410  IF R(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XR - X(1)) / (X(2) - X(1)):
     X(I) = XR: GOTO 360
420  IF T(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YT - Y(1)) / (Y(2) - Y(1)):
     Y(I) = YT: GOTO 360
430  IF B(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YB - Y(1)) / (Y(2) - Y(1)):
     Y(I) = YB: GOTO 360
```

比例法（ scaling ）

比例可由對角矩陣來決定。

$$(\ X\ ,\ Y\ )\begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix} = (\ AX\ ,\ DY\ )$$

利用對角矩陣可在個別的座標軸的方向上提供一獨立的比例。例如，我們以比例因素爲2向X方向伸展，由乘積得

$$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

執行程式4.5且令A＝2，B＝0，C＝0及D＝1

我們以因素爲1／5的比例向Y方向收縮，由乘積得

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

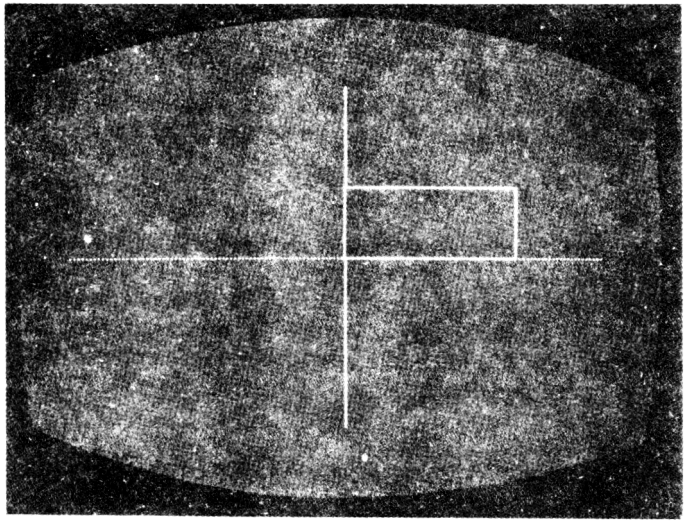
總比例矩陣是

$$\begin{pmatrix} K & 0 \\ 0 & K \end{pmatrix}$$

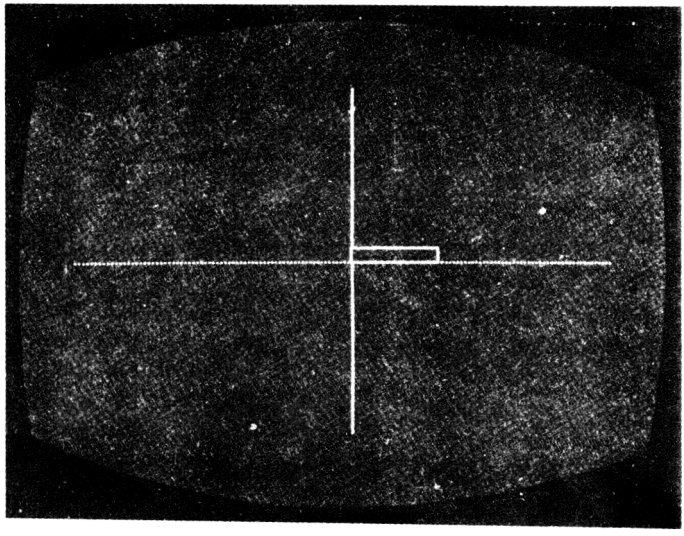
應用程式4.5，而改變A和D的值，固定B＝0和C＝0

對稱法（ reflection ）

對稱是指一物體經由X軸或Y軸產生於一個如物體照鏡子的像於各個軸的對邊的現象。用一個簡單的寫法來說明矩陣表示個別軸的對稱如下：



程式4.5的X軸比例乘2的比例變換



程式4.5的Y軸比例乘0.2的比例變換

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \text{和} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

用矩陣來表示經由原點（ 0 ，0 ）的變換矩陣爲

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

注意，對稱實質上主要也算是一種負的比例方法。

像的對稱亦可經由直線來形成，但是通常矩陣會不容易被決定。因此經由任意線的對稱處理在後面的“變換結合”運算一節中再與讀者一同討論。

切變法（ shear ）

程式 4.5 正方形的點切變其矩陣爲

$$\begin{pmatrix} 1 & 1.5 \\ 0 & 1 \end{pmatrix}$$

其代數式：

$$(X',Y') = (X,Y)\begin{pmatrix} 1 & 1.5 \\ 0 & 1 \end{pmatrix} = (X, 1.5X + Y)$$

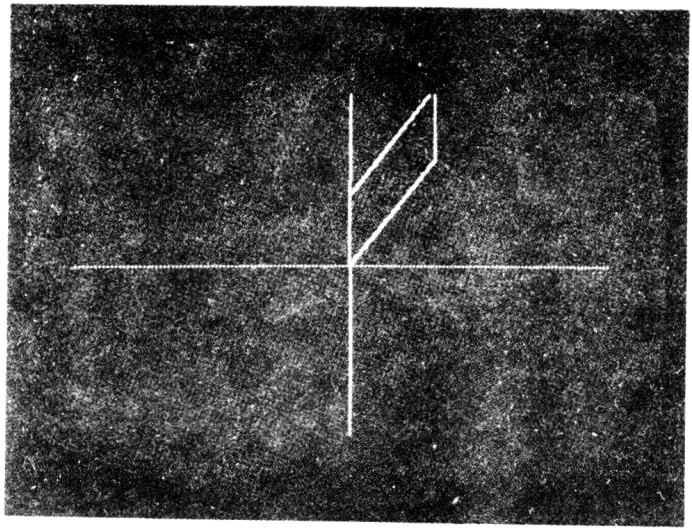
此變換是指示著Y的切變法（或切變於Y方向）；若用X切變法表示其矩陣型式：

$$\begin{pmatrix} 1 & 0 \\ C & 1 \end{pmatrix}$$

執行程式 4.5 且定義A，B，C和D之切變值

旋轉法（ rotation ）

欲使圖像的點依反時針方向旋轉時，可將點乘下面矩陣：



程式 4.5 的切變圖形

$$\begin{pmatrix} \text{COS}\theta & \text{SIN}\theta \\ -\text{SIN}\theta & \text{COS}\theta \end{pmatrix}$$

注意：符號 θ 是一希臘字母。許多數學書均用此字母代表角度，但在 BASIC 裏，我們用 TH 或 THETA 來代替 θ，而且我們也用括弧，所以 SINθ 在 BASIC 程式裏是如 SIN(TH) 之型式。

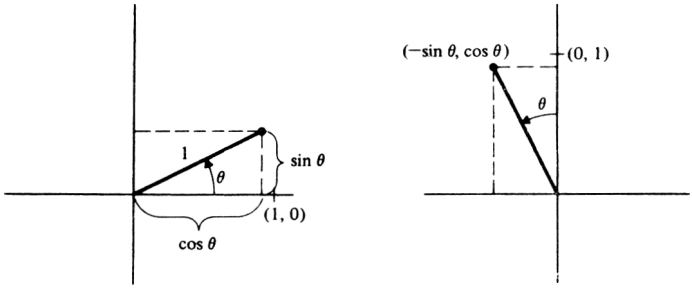


圖 4-7 點 (1,0) 及點 (0,1) 各變換結果

前面向讀者介紹的比例法、對稱法、和切變法所組成的結果是清楚且易判定。而旋轉法所組成的矩陣就顯然的會讓我們不容易接受。首先我們注意：

$$(A, B) = (1, 0)\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

和

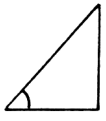
$$(C, D) = (0, 1)\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

因此第一列的變換矩陣說明了點（1，0）變換的結果；而第二列的變換矩陣說明了點（0，1）變換的結果。

從三角學的觀點來看，點（1，0）經反時針方向旋轉θ角後產生點（COSθ，SINθ），同理點（0，1）旋轉則產生點（$-$SINθ，COSθ）因此旋轉矩陣的表示法：

$$\begin{pmatrix} \text{COS}\theta & \text{SIN}\theta \\ -\text{SIN}\theta & \text{COS}\theta \end{pmatrix}$$

注意：SIN和COS對直角三角形邊長的比率關係：



$$\text{SIN}(\theta) = \frac{A}{C} \quad , \quad \text{COS}(\theta) = \frac{B}{C}$$

用此關係來印證上述旋轉的結果。

圖4.8說明了程式4.5中A = COS(.5) = .878，B = SIN(.5) = .479，C = $-$ SIN(.5) = $-$ .479和D = COS(.5) = .878的執行結果。

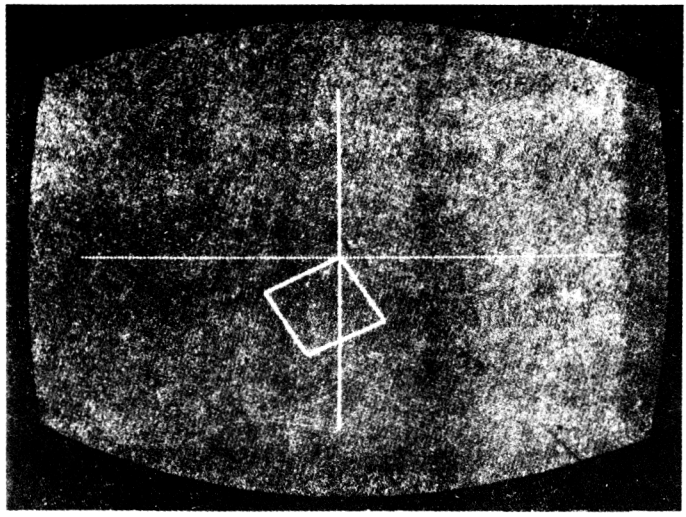


圖 4-8 程式 4.5 的旋轉圖形

注意：在BASIC 裏，角度須用弳度來測定。1 弳度大約 57.3 度，其正確關係是 2π 弳 = 360°，所以 1 弳度 = 360 / 6.2832 = 57.2956°。至於更詳細的說明請看附錄 5 。

雖然我們現在已了解下面的變換矩陣型態：本體法，比例法，對稱法，切變法及旋轉法。但是經驗告訴我們，使用旋轉法矩陣或以上現有的結合度變換矩陣的方式來處理被認明較正當及更較其它方法獲得便利。

舉例：定義變換 T 使其向反時針方向旋轉 θ = .4 弧度後再以 X 軸比例乘 2 作比例變換。

此旋轉後的點矩陣

$$(X', Y') = (X, Y)\begin{pmatrix} COS(.4) & SIN(.4) \\ -SIN(.4) & COS(.4) \end{pmatrix}$$

和及其比例變換得

$$(X'', Y'') = (X', Y')\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

(X', Y') 代入得

$$(X'', Y'') = \left[(X, Y)\begin{pmatrix} COS(.4) & SIN(.4) \\ -SIN(.4) & COS(.4) \end{pmatrix}\right]\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= (X, Y)\left[\begin{pmatrix} COS(.4) & SIN(.4) \\ -SIN(.4) & COS(.4) \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}\right]$$

$$= (X, Y)\begin{pmatrix} 2COS(.4) & SIN(.4) \\ -2SIN(.4) & COS(.4) \end{pmatrix}$$

因此T變換矩陣是：

$$\begin{pmatrix} 2COS(.4) & SIN(.4) \\ -2SIN(.4) & COS(.4) \end{pmatrix} \approx \begin{pmatrix} 1.842 & .389 \\ -.779 & .921 \end{pmatrix}$$

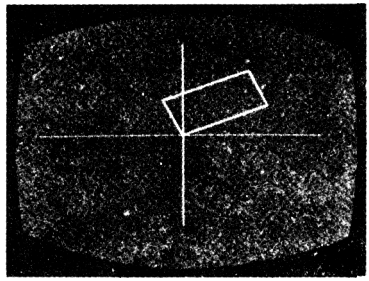
注意：在BASIC中2COS(.4)要寫成2*COS(.4)

連續地應用矩陣的變換來改變矩陣的乘積。當我們計算矩陣的結果時，必須很小心的注意這矩陣的正確順序。通常，矩陣乘積是不能倒置的（即 AB = BA 是不成立）。現給讀者證明矩陣不能倒置之計算，我們定一個與變換T不同的變換T* 我們先定T* 其X的比例變換

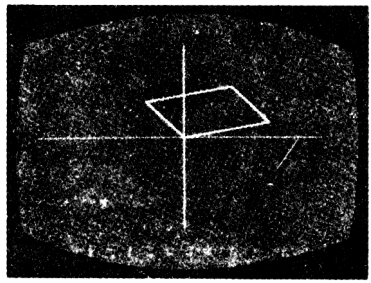
是2後再反時針方向旋轉θ = .4弳如此而求得變換T* 之乘積：

$$\begin{pmatrix} 1.842 & .779 \\ -.389 & .921 \end{pmatrix}$$

則T* 不同於T 。



程式4.5 T變換結果



程式4.5 T*變換結果

目前矩陣變換的方法有一個缺點，就是所有變換都以原點（0，0）爲中心，因此不論利用何者變換，原點（0，0）是永遠不變的，因此反時針旋轉是對原點（0，0）旋轉，不能對其它點旋轉；對稱對X軸，Y軸，或原點對稱，不能對其它線或其它點對稱。

爲了克服僅能以原點爲中心的變換現象，再向讀者介紹一些比較深入的變換法，以讓我們有更進一步的發現，期盼增加我們應用數學的觀念完成設計程式的簡化。

均質座標（Homogeneous Coordinates）

我們應了解平面參考系統裏點的表示通常採用二點座標的方法會顯得較便利及有效率。而參考座標軸彼此垂直相交於一個參考原始點，因此在此系統裏參考點（通常是原點）是常常先被設定的，且平面的每一點亦是從座標軸對映性的最短距離（垂直距離）的結合現象。

因此均質座標亦表示了點在平面系統的結合，只不過是將點（A，B）以次序性的（P，Q，R）表示。其中A＝P／R，B＝Q／R。

均質座標點表示法不是唯一性。例如點（2，5），可以表示爲（4，10，2），（－20，－50，－10），（2.1，5.25，1.05），或（2，5，1）與其它。通常我們使用（2，5，1）的表示法，因爲點（A，B）的標準均質座標是（A，B，1）。基於此，直角座標（P／R，Q／R）和標準化座標（P／R，Q／R，1）的均質座標均是（P，Q，R）。

當點（A，B）用標準均質座標（A，B，1）時，有時可以認爲那第三個座標是附加值，但事實上利用均質座標的方法可以增加我們對矩陣變換表示的技巧，讀者切勿疏忽。

向讀者介紹均質座標的概念其目的無非是想讓大家提高先前所介紹矩陣變換的表示技巧，進而明白下面矩陣變換的意義：

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \text{轉換} \quad \begin{pmatrix} A & B & 0 \\ C & D & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

平移（translation）

有了均質座標的先決概念後，我們加入平移法變換。所謂的“平移經由（H，K）”就是我們想要物體的點水平移動H單位和垂直移動K單位。雖然先前的平移並不能將矩陣表示出來，但是在均質座標

下却可表示：

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ H & K & 1 \end{pmatrix}$$

乘點（X，Y，1）則眞正平移的矩陣爲（X＋H，Y＋K，1）。

變換的結合 （Combinations Of Transformations）

與讀者討論了以上種種的矩陣後，不知您還滿意嗎？筆者知道有些變換是那些矩陣不能表示的。那是因爲某些變換要用數種結合來一同完成的，本節就與讀者討論結合變換的表示。例如，旋轉變換矩陣乃表示某一原點由反時針方向旋轉θ角，而現在我們欲得一個由點（20，30）向反時針方向旋轉30°的旋轉矩陣。

通常旋轉變換矩陣表示其旋轉體中心是在（0，0），而旋轉中心在（20，30）其結果是：

1. 移動點（20,30）到原點（平移原點到點（－20，－30）時之變換矩陣：

$$T=\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -20 & -30 & 1 \end{pmatrix}$$

2. 旋轉30°時變換矩陣：

$$R=\begin{pmatrix} COS30^\circ & SIN30^\circ & 0 \\ -SIN30^\circ & COS30^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$=\begin{pmatrix} .866 & .5 & 0 \\ -.5 & .866 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. 平移（0，0）到（20,30）。爲了恢復這系統的原始位置：

$$T^* = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 20 & 30 & 1 \end{pmatrix}$$

因此點（20,30）旋轉30°後至（X′，Y′）的均質座標爲（X′，Y′,1）=（X，Y，1）TRT* 。我們首先求得M = TRT* 再求（X′，Y′，1）=（X，Y，1）M，而

$$M = \begin{pmatrix} .866 & .5 & 0 \\ -.5 & .866 & 0 \\ 17.68 & -5.98 & 1 \end{pmatrix}$$

及（X′,Y′,1）=（.866X − .5Y + 17.68，5X + .866Y − 5.98，1）。

致使了均質座標的變換矩陣的型式是：

$$\begin{pmatrix} A & B & 0 \\ C & D & 0 \\ H & K & 1 \end{pmatrix}$$

和描述點（X，Y）的變換結果爲

$$(X',Y',1) = (X,Y,1)\begin{pmatrix} A & B & 0 \\ C & D & 0 \\ H & K & 1 \end{pmatrix}$$

或X′ = AX + CY + H和Y′ = BX + DY + K 。

程式4.6爲程式4.5之修改式，而可自行提供A，B，C，D，H和K值，而讓我們得知此型變換的結果。圖4.10 爲均質座標變換的圖形。

程式4·6：均質座標

```
1  REM PROGRAM 4.6 (HOMOGENEOUS COORDINATES)
2  REM ILLUSTRATES EFFECT OF 3x3 TRANSFORMATION
   MATRIX
10 CX = 140:CY = 96:SC = 1.16
20 XR = 279:XL = 0:YT = 0:YB = 191
```

```
30  TEXT : HOME
40  INPUT " A = ";A
50  INPUT " B = ";B
60  INPUT " C = ";C
70  INPUT " D = ";D
80  INPUT " H = ";H
90  INPUT " K = ";K
100  HGR : HCOLOR= 6
110  HPLOT 0,96 TO 279,96: HPLOT 140,0
     TO 140,191
120  HCOLOR= 3
130  FOR S = 1 TO 4
140  FOR I = 1 TO 2
150  READ X,Y
160 TX = A * X + C * Y + H:
    TY = B * X + D * Y + K
170 X(I) = SC * TX + CX:Y(I) = CY - TY
180  GOSUB 320
190  NEXT I
200  GOSUB 400
210  NEXT S: RESTORE
220  VTAB 21: PRINT "A  B"; TAB( 11);A; TAB
     ( 20); B
230  VTAB 22: PRINT "C  D"; TAB( 8);"="; TAB
     ( 11);C; TAB( 20);D
240  VTAB 23: PRINT "H  K"; TAB( 11);H; TAB
     ( 20);K
250  VTAB 24: PRINT "PRESS 'N' FOR NEW MATRIX;
     'ESC' TO END";
260  VTAB 10: GET A$: IF A$ = "N" THEN 30
270  IF ASC (A$) = 27 THEN TEXT : HOME : END
280  GOTO 260
290  DATA 0,0,40,0,40,0,40,40,40,40,0,40,0,40, 0,0
300  END
310  REM **** CLASSIFY SUBROUTINE
320 L(I) = 0:R(I) = 0:T(I) = 0:B(I) = 0
330  IF X(I) < XL THEN L(I) = 1
340  IF X(I) > XR THEN R(I) = 1
350  IF Y(I) < YT THEN T(I) = 1
360  IF Y(I) > YB THEN B(I) = 1
370  RETURN
380  GOSUB 320
390  REM **** CLIPPING SUBROUTINE
400  IF L(1) * L(2) + R(1) * R(2) + T(1) *
     T(2) + B(1) * B(2) < > 0 THEN RETURN
410 I = 1: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN I = 2: IF L(I) + R(I) + T(I) + B(I) = 0
    THEN HPLOT X(1),Y(1) TO X(2),Y(2): RETURN
420  IF L(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XL - X(1)) / (X(2) - X(1)):X(I) =
     XL: GOTO 380
```

(continued)

```
430  IF R(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XR - X(1)) / (X(2) - X(1)):X(I) =
     XR: GOTO 380
440  IF T(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YT - Y(1)) / (Y(2) - Y(1)):
     Y(I) = YT: GOTO 380
450  IF B(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YB - Y(1)) / (Y(2) - Y(1)):Y(I) =
     YB: GOTO 380
```

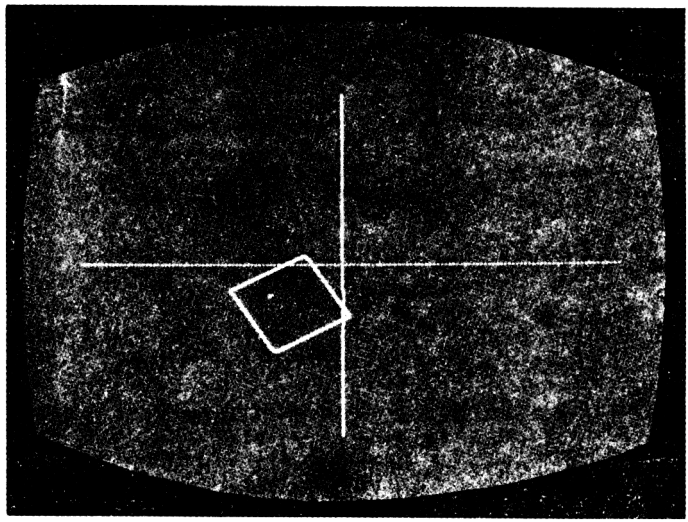


圖 4-10　均質座標變換情形

利用變換矩陣，我們再擴大如圖 4.11 所示的六邊形巢串的程式。首先我們須定六邊形的中點在（ 0 ，0 ）及先決定其中一個頂點。由於六邊形的六個頂點是以每中心 60 度而分離。因此第二頂點是從第一頂點旋轉60° 而得。第三頂點亦是第二頂點旋轉 60° 而得。如此類推，獲得了第 4 ，5 及 6 的頂點。旋轉變換矩陣是

$$R = \begin{pmatrix} COS\theta & SIN\theta & 0 \\ -SIN\theta & COS\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

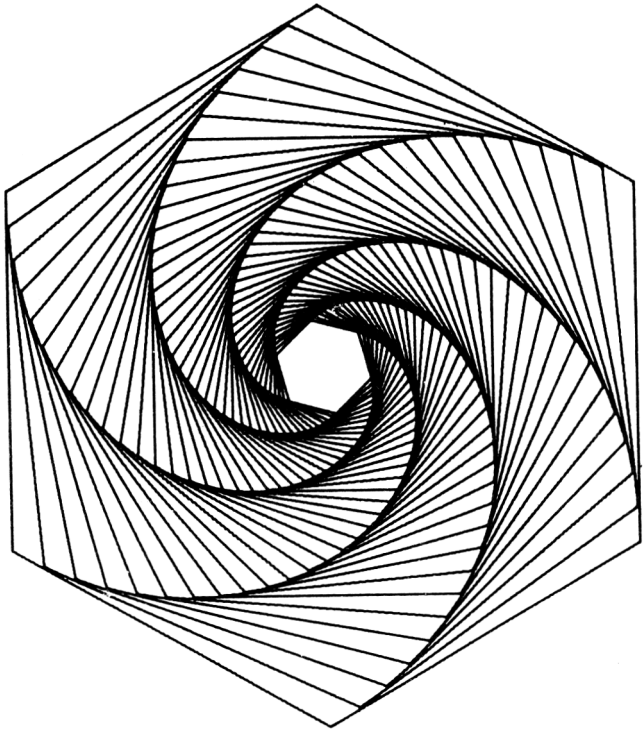


圖 4-11 巢串六邊形

在程式計算的步驟中，會由目前所得的頂點座標，以（ XN ， YN ，1 ）=（ X ，Y ，1 ）R而求得下一次頂點的座標（ XN ，YN ），故此我們設計此兩點是：

```
XN = X*COS(TH) - Y*SIN(TH)
YN = X*SIN(TH) + Y*COS(TH)
```

因爲要計算 SIN和COS所須時間較長，所以我們可以先定C = COS（60°）和S = SIN(60°)，則（ XN ，YN ）的計算成：

```
XN = X*C-Y*S
YN = X*S+Y*C
```

又因我們當初爲了方便旋轉變換所以把六邊形中點定在（ 0 ，0 ）處

，但對 Apple Ⅱ顯示幕會發生困難，原因是會造成大部份的頂點落在顯示幕之外，如圖 4.12

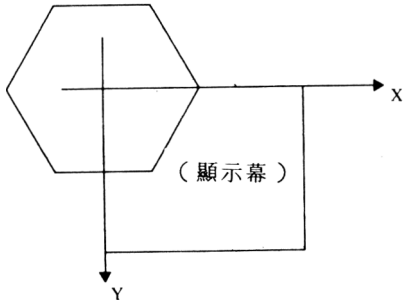


圖 4-12　Apple II 顯示幕

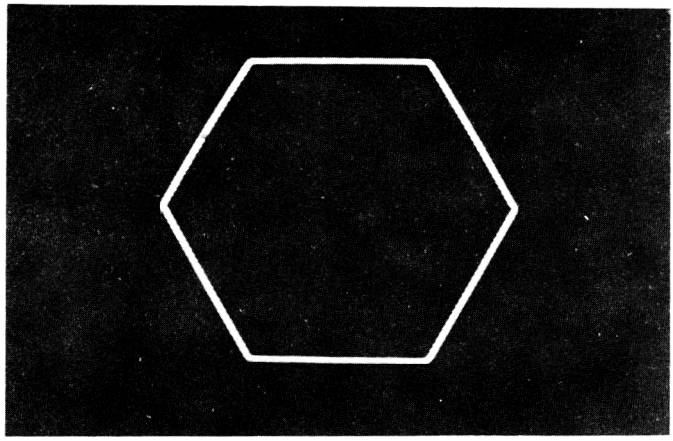
基於此原因在繪六邊形點之前，必須先將圖平移至顯示幕位置之內，則此點移動至（ CX, CY ）的平移矩陣

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ CX & CY & 1 \end{pmatrix}$$

下面的程式我們可以繪單一的六邊形：

程式 4·7 ：六邊形

```
1  REM PROGRAM 4.7 (HEXAGON)
2  REM DRAWS A SINGLE HEXAGON
10 PI = 3.14159
20 C = COS (PI / 3):S = SIN (PI/3)
40 X = 95:Y = 0:CX = 140:CY = 96:SC = 1.16
50  HGR2 : HCOLOR= 3
70  FOR I = 0 TO 6
80 SX = X * SC + CX:SY = CY + Y
90  IF I = 0 THEN HPLOT SX,SY
100  HPLOT TO SX,SY
110 XN = X * C - Y * S:Y = X * S + Y *
    C:X = XN
120  NEXT I
```



我們可再加一些巢串敍述於程式4.7，以提供繪出陸陸續續的六邊形，每一新六邊形均比前一六邊形稍小及比前一六邊形稍微旋轉。

可用矩陣的乘積來決定六邊形的大小：

$$\begin{pmatrix} SF & 0 & 0 \\ 0 & SF & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

此處的SF是我們自訂的比例因素（ scale factor），我們再用旋轉變換矩陣來爲每一稍爲旋轉的每個六邊形定一個新旋轉θ值。

下面的轉移表示是從上一個的六邊形（包括比例變換，旋轉變化）到下一個六邊形的變化，如下：

$$(XN,YN,1)=(X,Y,1)\begin{pmatrix} SF & 0 & 0 \\ 0 & SF & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} COS\theta & SIN\theta & 0 \\ -SIN\theta & COS\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

或

```
XN = X*SF*COS(TH) - Y*SF*SIN(TH)
YN = X*SF*SIN(TH) + Y*SF*COS(TH)
```

以上的觀念建立之後，我們可使用程式4.8來繪圖了。

注意和建議

1. 模擬程式4.7的作法來設計五邊形或八邊形。或修改程式使它能隨操作者之意畫任N邊形。

2. 寫一個顯示英文姓名的首字母於繪圖幕的程式而利用指定的變換來執行。

(a)顯示在三種位置（用平移法）。

(b)旋轉45°（用旋轉法）。

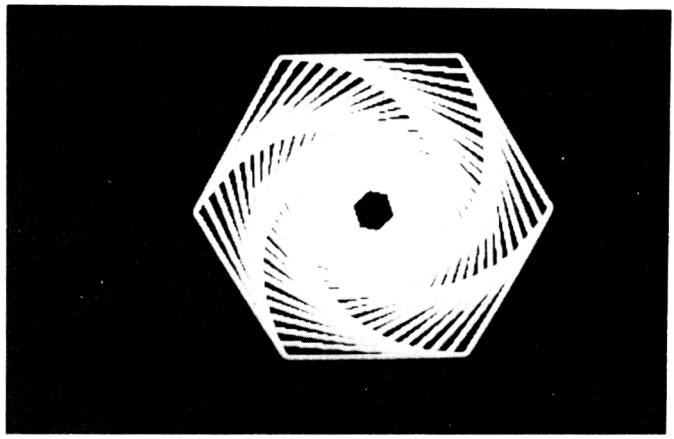
(c)字母上下反置（用對稱法）。

(d)使用三種尺寸（用比例法）。

3. 在顯示幕上設計一個圖案（如程式4.8），利用觀測台的安排

程式4·8：巢串六邊形

```
 1  REM PROGRAM 4.8 (HEXAGON DESIGN)
 2  REM DRAWS A SEQUENCE OF SPIRALLING HEXAGONS
10 PI = 3.14159
20 C = COS (PI / 3):S = SIN (PI / 3)
30 C1 = COS (PI / 36):S1 = SIN (PI / 36):SF =
   .95
40 X = 95:Y = 0:CX = 140:CY = 96:SC = 1.16
50  HGR2 : HCOLOR= 3
60  FOR J = 1 TO 40
70  FOR I = 0 TO 6
80 SX = X * SC + CX:SY = CY + Y
90  IF I = 0 THEN HPLOT SX,SY
100  HPLOT TO SX,SY
110 XN = X * C - Y * S:Y = X * S + Y * C:X = XN
120  NEXT I
130 XN = SF * (X * C1 - Y * S1):Y = SF *
    (X * S1 + Y * C1):X = XN
140  NEXT J
```



而顯示（彼此間不可擦掉）。

(a)觀測台顯示在右上角部份，設定

（ XR = 250 ， XL = 40 ， YT = 5 ， YB = 80 ）

(b)觀測台顯示在左下角部份，設定

（ XR = 130 ， XL = 90 ， YT = 90 ， YB = 150 ）

第五章　解析幾何與電腦繪圖

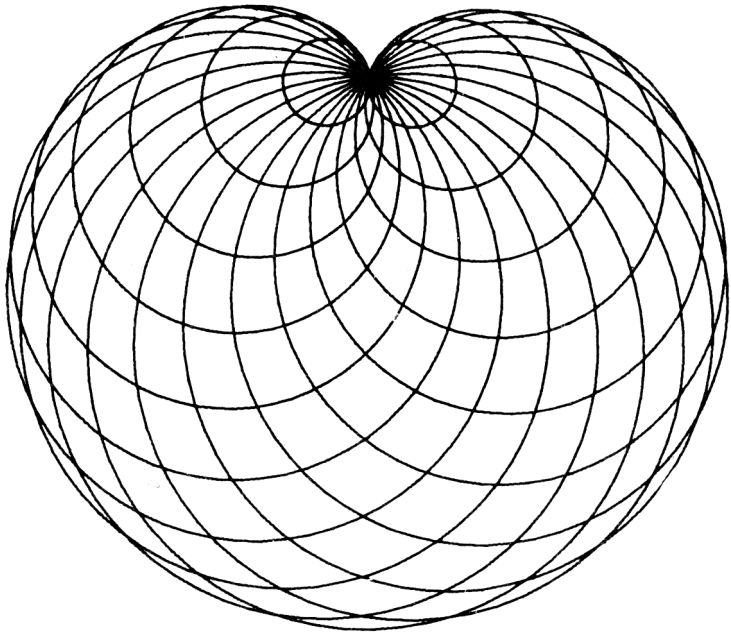


圖 5 -1

在今天數學圖形的討論已是高中學校課程中的重要話題之一，它不僅是解析幾何的一部份，亦是要研究代數學（或解析學）和製圖學二者間數學關係的起點。

本章的前面裏我們進行解析幾何的模擬，暫時將重點放在設計，

而不是電腦的計算及繪圖。

座標軸（Coordinate Axes）

解析幾何通常也叫“座標幾何”因爲它完全是依賴著座標軸系統。而學生們在開始作幾何分析時也就是直接畫相互垂直相交於中點的兩條線，如圖5.2

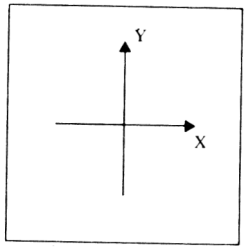


圖 5-2 幾何分析的二條座標軸

而 Apple Ⅱ高解像度顯示幕所顯示的座標軸系統與先前所提的座標軸系統有少許不同。Apple 座標系統如圖5.3

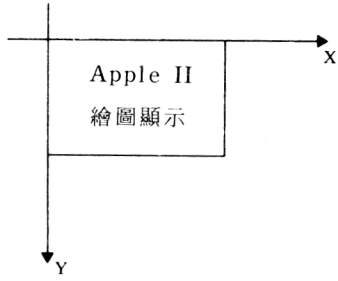


圖 5-3 Apple 座標系統

但讀者仍須適應及重新建立各點的位置，因此我們要模擬是在向上爲Y，向右爲X的座標系統中求出相對的中央點的位置。

首先，在座標軸上定出二條模擬線

```
HPLOT 140,0 TO 140,191: HPLOT 0,96 TO 279,96
```

因此座標系統之中央點：

```
CX=140: CY=96
```

現在我們爲了以後寫程式方便更換 HPLOT X,Y 成 HPLOT X + CX，CY − Y 。此時點（ X ，Y ）的有意義範圍是

$$-140 \leq X \leq 139 \quad 及 \quad -96 \leq Y \leq 95$$

了解了上者後，告訴讀者這種的方便其實是做了一種變換：X 軸的對稱法及再對點（ CX ，CY ）的平移法 。

對稱矩陣是

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

和平移矩陣是

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ CX & CY & 1 \end{pmatrix}$$

矩陣的乘積是

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ CX & CY & 1 \end{pmatrix}$$

及

$$(X,Y,1)\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ CX & CY & 1 \end{pmatrix} = (X+CX, CY-Y, 1)$$

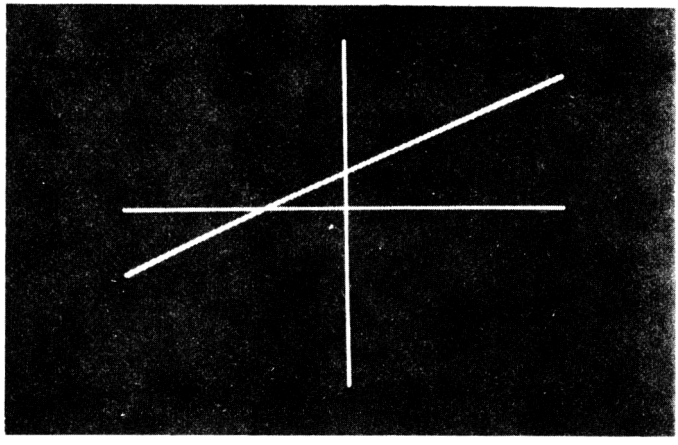
因此只要定 X 和 Y 值，計算 SX = X + CX 及 SY = CY − Y 則我們就可以在此中央座標系統中顯示幕的相對位置（ SX,SY ）。若想使用比例因素 SC 就 SX = SC*X + CX 及 SY = CY − Y。在本章的程式中大部份都採用後者的型式 。

直線（Lines）

除了點之外，幾何圖形上的另一個主角是直線，直線代數上的表示是Y＝MX＋B。其中M是直線的斜率（slope），用來測定線的陡度；B是直線相交於Y軸時Y點截距。程式5.1爲一畫Y＝.4X＋20圖於座標系統如下

程式5·1：直線

```
1   REM PROGRAM 5.1 (LINES)
2   REM LINE Y = MX+B
10   HGR2 : HCOLOR= 2
20   HPLOT 140,0 TO 140,191: HPLOT 0,96 TO 279,96
30   HCOLOR= 3:CX = 140:CY = 96
40 X = -140:Y = .4 * X + 20
50 SX = X + CX:SY = CY - Y
60   HPLOT SX,SY
70 X = 139:Y = .4 * X + 20
80 SX = X + CX:SY = CY - Y
90   HPLOT TO SX,SY
```



程式5.1所呈現的直線應是毫無問題，但仍因沒有使用比例因素

SC，因此或許呈現的是不正確的斜率。此外，如果直線 Y = 2X + 20 時若 X = − 140 及 X = 139 則點會座落於顯示幕之外，故須做必要的截割，至於截割處理在第四章已討論過在此不多提，而下面線截割的程式加入了程式 4.1 基本部份後而成如下：

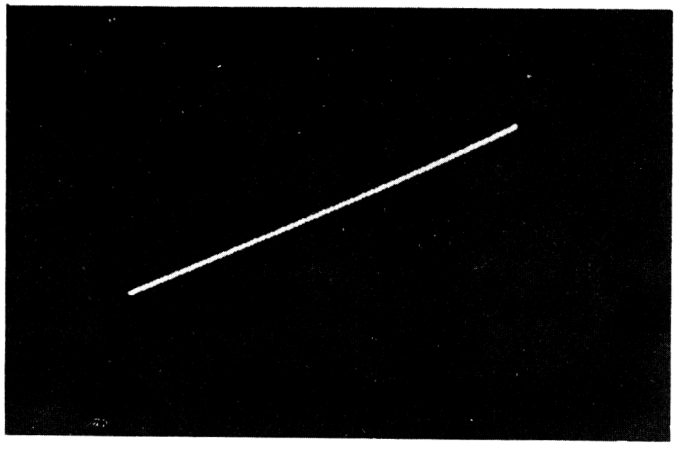
程式 5·2：直線的截割

```
1   REM PROGRAM 5.2 (CLIPPED LINE)
2   REM ILLUSTRATES USE OF LINE CLIPPING
    SUBROUTINE
10  XR = 279:XL = 0:YT = 0:YB = 191
20   HGR2 : HCOLOR = 2:SC = 1.16
30   HPLOT 140,0 TO 140,191: HPLOT 0,96 TO 279,96
40   HCOLOR= 3:CX = 140:CY = 96
50  I = 1:X = - 140:Y = .4 * X + 20:X(I) = SC * X
    + CX: Y(I) = CY - Y: GOSUB 100
60  I = 2:X = 139:Y = .4 * X + 20:X(I) = SC * X +
    CX: Y(I) = CY - Y: GOSUB 100
70   GOSUB 200
80   END
95   REM **** CLASSIFY SUBROUTINE
100 L(I) = 0:R(I) = 0:T(I) = 0:B(I) = 0
110  IF X(I) < XL THEN L(I) = 1
120  IF X(I) > XR THEN R(I) = 1
130  IF Y(I) < YT THEN T(I) = 1
140  IF Y(I) > YB THEN B(I) = 1
150  RETURN
190  GOSUB 100
195  REM **** CLIPPING SUBROUTINE
200  IF L(1) * L(2) + R(1) * R(2) + T(1) * T(2)
     + B(1) * B(2) < > 0 THEN RETURN
210 I = 1: IF L(I) + R(I) + T(I) + B(I) = 0 THEN
    I = 2: IF L(I) + R(I) + T(I) + B(I) = 0 THEN
    HPLOT X(1),Y(1) TO X(2),Y(2): RETURN
220  IF L(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XL - X(1)) / (X(2) - X(1)):X(I) =
     XL: GOTO 190
230  IF R(I) = 1 THEN Y(I) = Y(1) + (Y(2) -
     Y(1)) * (XR - X(1)) / (X(2) - X(1)):X(I) =
     XR: GOTO 190
240  IF T(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YT - Y(1)) / (Y(2) - Y(1)):Y(I) =
     YT: GOTO 190
250  IF B(I) = 1 THEN X(I) = X(1) + (X(2) -
     X(1)) * (YB - Y(1)) / (Y(2) - Y(1)):Y(I) =
     YB: GOTO 190
```

如此說來，直線的繪圖考慮的問題似乎已不必再深入討論了。原因是直線的代數式是 Y = MX + B ，而我們在顯示幕的可見部份求出點再利用 HPLOT X1,Y1 TO X2,Y2 的命令，即可很快及容易地繪出此條線來。然而，隨設計者的需要，有時候要直線緩慢地呈現或說欲某物體沿著直線的軌跡移動時，我們就有必要對直線上的每點加以考慮。程式 5.3 或許可以說明此現象。

程式 5·3：直線變量

```
1   REM PROGRAM 5.3 (LINE)
2   REM DRAWS LINE SLOWLY
10   HGR2 : HCOLOR= 3
20 X1 = 10:Y1 = 150:X2 = 250:Y2 = 50
30 DX = X2 - X1:DY = Y2 - Y1
40   HPLOT X1,Y1
50   FOR T = 0 TO 1 STEP .001
60   HPLOT TO X1 + T * DX,Y1 + T * DY
70   NEXT T
```



本程式所執行的直線是從（X1，Y1）以 X 變量 DX = X2 − X1 和 Y 變量 DY = Y2 − Y1 而畫到（X2，Y2 ）。行號 50 到 70 是分成

1000次小變量的累增迴路（ loop ），使得直線的呈現才不至於猛然發生（如HPLOT X1,Y1 TO X2,Y2 ），而且累增迴路的增量也控制直線的移動率。

圓（Circles）

圓是從圓心（H，K）到一定距離（R）的所有點（X，Y）的集合。圓的代數式是$(X-H)^2+(Y-K)^2=R^2$。因此若圓心在（0，0）則等式

$$X^2+Y^2=R^2 \qquad (式5.1)$$

有很多種方法可以讓我們利用X和Y的關係在繪圖幕上產生圓的圖像。下面介紹三種簡單的方法來產生圓。

直角座標法（Rectangular Coordinate Method）

$X^2+Y^2=R^2$ 可分解成兩個函數式 $YR=\sqrt{R^2-X^2}$ 及 $YN=-\sqrt{R^2-X^2}$。程式5.4即利用此二函數來產生 $X^2+Y^2=8100$ 的圓。

程式5·4：圓之一

```
1   REM PROGRAM 5.4 (CIRCLE 1)
2   REM RECTANGULAR COORDINATE CIRCLE
10  CX = 140:CY = 96
20   DEF FN YP(X) = SQR (8100 - X * X)
30   DEF FN YN(X) = -SQR (8100 - X * X)
40   HGR2 : HCOLOR= 3
50  X = 90:Y = 0:SX = X + CX:SY = CY - Y
60   HPLOT SX,SY
70   FOR X = 80 TO -90 STEP -10
80  Y = FN YP(X)
90  SX = X + CX:SY = CY - Y
100  HPLOT TO SX,SY
110  NEXT X
120  FOR X = - 80 TO 90 STEP 10
130 Y = FN YN(X)
140 SX = X + CX:SY = CY - Y
150  HPLOT TO SX,SY
```

```
160  NEXT X
170  END
```

而程式 5.4所顯示的像有兩點將值得與讀者一起討論：

1. 比例大小不同：它不呈現“正圓”，其原因是Apple高解像顯示幕的限制（本書第四章曾討論）。補救的辦法是在點（X，Y）的X座軸上先乘上比例因素 SF 後再繪圖。
2. 圓周不圓滑：此程式是利用行號50～110 顯示上半圓；行號120～160 顯示下半圓，則此兩半圓兩端連接點上產生了突狀。補救的辦法是：使畫點迴路的步增值（ step ）設定小一點，使得產生數量密集的點。如此即可改善先前不圓滑的現象。但不好，由於如此一來，在執行半圓中間部份時，因爲此部份原本就已圓滑不必再修飾，而徒耗時間在這裏，使得速度變慢很多。

由於突狀的產生乃由於畫點迴路一直以固定的X軸增量值作爲點與點間的距離，這樣在圓弧的部份即出現不勻稱的突狀現象。改行號100與行號150讀者就更能看清楚此不勻稱的現象。

```
100 HPLOT SX,SY
150 HPLOT SX,SY
```

角度增量法 （Angular Increments Method）

利用程式5.4的結果由於計算點不是以一個均勻間隔來圍成一個圓弧以致產生一個突狀的圓像，因此我們在這裏利用本法的技巧來設計一個相等間隔的點距離。

從三角學的觀點來看，圓 $X^2 + Y^2 = R^2$ 上的點（X，Y）可表示成

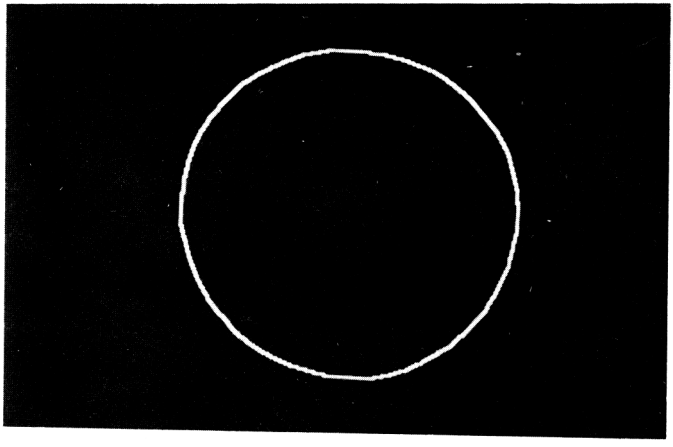
$$X = RCOS(\theta)$$
$$Y = RSIN(\theta) \qquad \text{（式5.2）}$$

而θ爲圖5.4所示。

程式5.5利用了這公式來畫一個相等小間隔點的圓：

程式5·5：圓之二

```
1  REM PROGRAM 5.5 (CIRCLE 2)
2  REM DRAWS CIRCLE USING ANGULAR INCREMENTS
10 CX = 140:CY = 96:SC = 1.16:FL = 0:R = 90
20  HGR2: HCOLOR= 3
30  FOR T = 0 TO 6.4 STEP .3
40 X = R * COS (T):Y = R * SIN (T)
50 SX = SC * X + CX:SY = CY - Y
60  IF FL = 1 THEN 80
70  HPLOT SX,SY:FL = 1
80  HPLOT TO SX,SY
90  NEXT T
```



任選的SC乃視使用的設備而定（見第4章）。

我們可以使用命令HPLOT TO來繪出線段，但必須在此命令之前已由HPLOT或已繪出線段的HPLOT TO的命令定出起始點後方可使用。程式5.4即提供了一個最初點於行號60的HPLOT，而程式5.5則使用了一個標幟（flag）（FL）來定有（FL = 1）或沒有（FL = 0）提供了起始點，如此即可利用迴路來產生點而引導全部的繪圖。

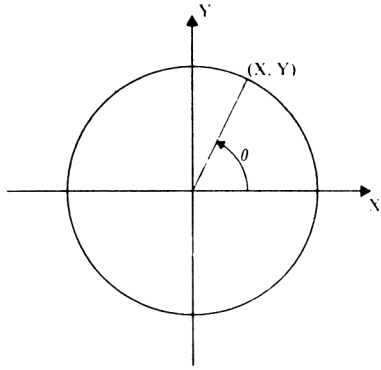


圖 5-4

點重覆法 （ Recursive Of Points Method ）

程式 1.7時曾畫過圓，當時就是使用這種方法，並且計算點的技巧也較前迅速。

重複計算SIN(θ)及COS(θ) 的值是相當費時的，所以在這裏我們要求對於SIN和COS的處理只要計算一次。

我們須計算利用產生的點來繪圖樣。例如程式 5.5就被計算出 22個點如（ X_1 , Y_1 ），（ X_2 , Y_2 ），（ X_3 , Y_3 ）……（ X_{22} , Y_{22} ），而每兩個連續點（ X_n , Y_n ）及（ X_{n+1} , Y_{n+1} ）可以表爲：

$$X_n = RCOS(\theta)$$
$$Y_n = RSIN(\theta)$$
$$X_{n+1} = RCOS(\theta + D\theta)$$
$$Y_{n+1} = RSIN(\theta + D\theta)$$

Dθ 值爲產生點廻路時的廻路增量。

從三角學中得知

$$X_{n+1} = RCOS(\theta + D\theta)$$
$$= RCOS(\theta)COS(D\theta) - RSIN(\theta)SIN(D\theta)$$

$$= X_n COS(D\theta) - Y_n SIN(D\theta)$$

$$Y_{n+1} = RSIN(\theta + D\theta)$$

$$= RSIN(\theta)COS(D\theta) + RCOS(\theta)SIN(D\theta)$$

$$= Y_n COS(D\theta) + X_n SIN(D\theta)$$

如此由於計算COS(θ)和SIN(θ)一次得知X_n和Y_n的值，再使用上面等式輕易得知X_{n+1}和Y_{n+1}。合併等式則我們的程式改寫成（將Dθ用DT，COS(Dθ)用C，SIN(Dθ)用S來代替）

```
T = X*C - Y*S: Y = Y*C + X*S: X=T
```

程式5.3即說明了上式用變數T暫時代替變數X後再求新的Y值計算。

至目前爲止，每個產生圓的程式其定圓的中心都在（0，0），我們也可能使中心點在（H，K）處，則繪圖時將其平移，點（X，Y）可平移至（X＋H，Y＋K）其矩陣乘積：

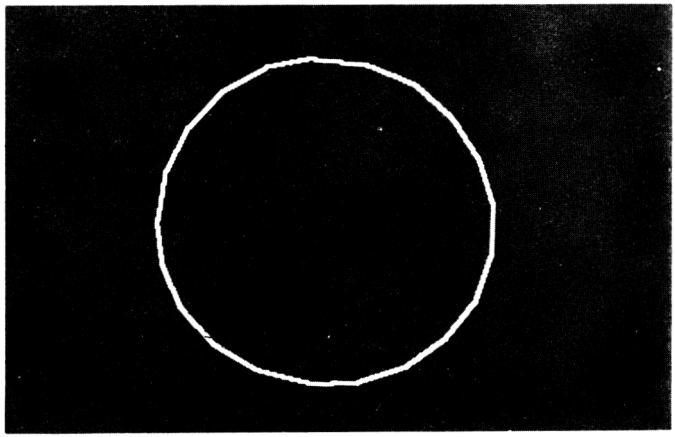
$$(X, Y, 1)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ H & K & 1 \end{pmatrix} = (X+H, Y+K, 1)$$

我們更換行號60的敍述，使程式包括了平移：

```
60 SX = SC*(X+H) + CX: SY = CY - (Y+K)
```

程式5·6：圓之三

```
1   REM PROGRAM 5.6 (CIRCLE 3)
2   REM RECURSIVELY CALCULATES POINTS ON A CIRCLE
10  CX = 140:CY = 96:SC = 1.16:X = 90:Y = 0:FL = 0
20  DT = .1:C = COS (DT):S = SIN (DT):N = 6.4 / DT
30   HGR2 : HCOLOR= 3
40   FOR I = 1 TO N
50  T = X * C - Y * S:Y = Y * C + X * S:X = T
60  SX = SC * X + CX:SY = CY - Y
70   IF FL = 1 THEN 90
80   HPLOT SX,SY:FL = 1
90   HPLOT TO SX,SY
100   NEXT I
```



截 割

圓的平移或因所選圓的半徑值太大都有可能影響點在計算時超出顯示幕的位置。這時我們就得把第四章開始所講的截割慣例應用在此時產生圓的程式了。不過，由於點很多，則欲測每兩小點的線段與顯示幕界的交點是一件很耗時的方法。因此我們利用程式5.7的技巧，採用只要線段有一個或一個以上之端點在顯示幕界外，即將線段捨棄，以節省繪圖時間的消耗。程式5.6之修改程式如下：

程式5·7：圓的截割

```
1  REM PROGRAM 5.7 (CLIPPED CIRCLE)
2  REM DRAWS A TRANSLATED CIRCLE
10 H = -40:K = 20
20 CX = 140:CY = 96:SC = 1.16:X = 90:Y = 0:FL = 0
30 DT = .1:C = COS (DT):S = SIN (DT):N = 6.4 / DT
40  HGR2 : HCOLOR= 3
50  FOR I = 1 TO N
60 T = X * C - Y * S:Y = Y * C + X * S:X = T
70 SX = SC * (X + H) + CX:SY = CY - (Y + K)
80  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 120
```

```
90  IF FL = 1 THEN 110
100  HPLOT SX,SY:FL = 1
110  HPLOT TO SX,SY
120  NEXT I
```

行號80 乃根據假如遭遇到超出顯示幕的點則停止劃而發生截割作用。注意 FL 是被重置在“0”所以剛開始這點在顯示幕上結果用HPLOT而不是HPLOT TO執行。

橢圓（Ellipses）

小學生他們在上美術課或算術課時常常使用一隻筆、二顆小圖釘、及一根細線就可很輕易的畫出軌跡圓滑的橢圓圖形，這個方法的程序如下：

1 使用圖釘，將細線兩端釘牢在紙張或紙板上，注意細繩仍須鬆弛，不可緊繫。

圖 5-5

2 用鉛筆芯緊拉細線定於一點。

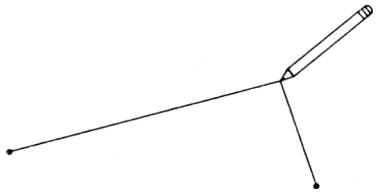


圖 5-6

3. 移動筆芯與細線的緊拉關係，從始點沿著緊拉軌跡畫圖至終點，此圖即是橢圓

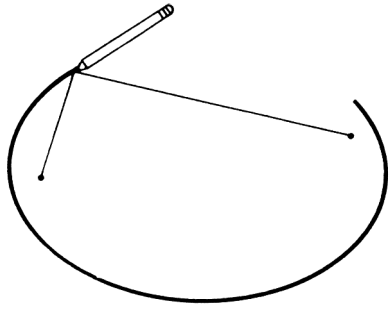


圖 5-7

由此可知橢圓的形成仍依二個圖釘的位置及細線長度而定（其實細線的長度與圖釘間的位置之比是學生們解析幾何重要的數據）。

由橢圓的聯想使我們想知道圖釘與細線間的關係如何：

橢圓是任一點到兩定點（圖釘）間之距離和爲常數（細線）的所有點集合。而固定的兩點叫做橢圓的“焦點”。相信大家定對這令人喜愛曲線的形成感興趣，讓我們來對一個我們想要的大小、形狀（上下或左右）、及位置的橢圓下定義吧。

如圖5.8所示，習慣上規定C點座標（中心點）爲（H，K）；AC（＝BC）的距離我們標明爲A；EC（＝CD）標明爲B（提示：在AB線段上的焦點（圖釘）至中心點的距離是$\sqrt{A^2-B^2}$。而細線長是2A）利用以上的標明符號，橢圓的代數式：

$$\frac{(X-H)^2}{A^2}+\frac{(X-K)^2}{B^2}=1 \qquad (式5.3)$$

雖然這橢圓方程式的型式是最常見及可正確的表示橢圓的所有點，但是有其它的方程式却比上面的更便利，及更有效率，在此筆者願再與讀者共同討論橢圓。

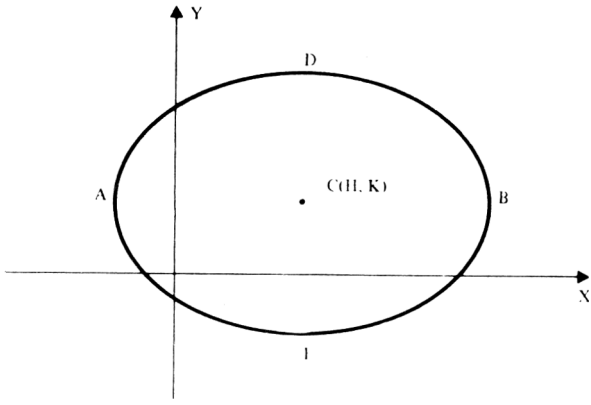


圖 5-8　完整橢圓

中心點（０，０）的橢圓

我們簡化以上的代數，爲使橢圓的中心點在（０，０）處（應用平移變換很輕易地定新位置）式 5.3 改換成

$$\frac{X^2}{A^2}+\frac{Y^2}{B^2}=1 \qquad (式 5.4)$$

使用上式畫橢圓的結果，使我們聯想到用 $X^2+Y^2=R^2$ 表示圓的現象——若是畫得太大則將不會構成圓滑的圖像；若是點數量不多，亦會呈現突狀的圖像。

將 5.4 式的代數式改成如下：

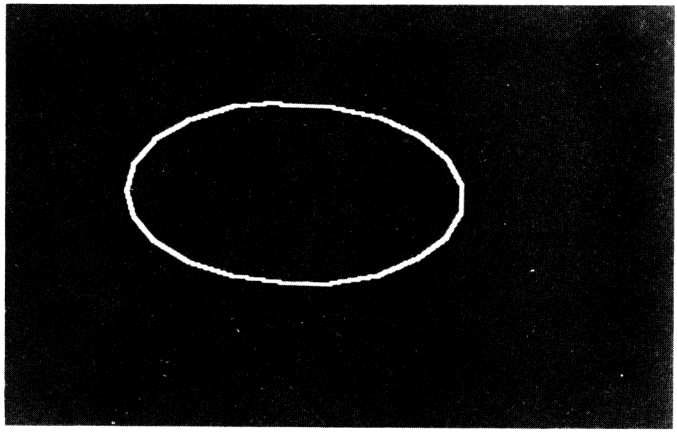
$$X = ACOS(\theta)$$

$$Y = BSIN(\theta)\ ,\ for\ 0 \le \theta \le 2\pi \quad (式 5.5)$$

而我們就利用 5.5 式的觀念設計出程式 5.8。

程式 5·8：橢圓之一

```
1  REM PROGRAM 5.8 (ELLIPSE 1)
2  REM DRAWS AN ELLIPSE CENTERED ON SCREEN
10 CX = 140:CY = 96:SC = 1.16:FL = 0:A = 90:
   B = 50
20  HGR2 : HCOLOR= 3
30  FOR TH = 0 TO 6.4 STEP .3
40 X = A * COS (TH):Y = B * SIN (TH)
50 SX = SC * X + CX:SY = CY - Y
60  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 100
70  IF FL = 1 THEN 90
80  HPLOT SX,SY:FL = 1
90  HPLOT TO SX,SY
100  NEXT TH
```



平移和旋轉(translation and rotation)

改變程式 5.8中 A 和B 之值，來看橢圓的變化。由於橢圓的方向與位置已被限制，所以無論如何變化，橢圓中心點總在(0 , 0)，橢圓的軸不是平行水平座標軸就是平行垂直座標軸。讓我們再增加橢圓旋轉和平移的變化吧 。

點旋轉 θ 角度的變換矩陣表示式

$$\begin{pmatrix} COS(\theta) & SIN(\theta) & 0 \\ -SIN(\theta) & COS(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

而平移至（H，K）的平移矩陣

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ H & K & 1 \end{pmatrix}$$

因此從一個圖像旋轉θ角後，再平移至（H，K）之矩陣表示式爲

$$\begin{pmatrix} COS(\theta) & SIN(\theta) & 0 \\ -SIN(\theta) & COS(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ H & K & 1 \end{pmatrix}$$

$$= \begin{pmatrix} COS(\theta) & SIN(\theta) & 0 \\ -SIN(\theta) & COS(\theta) & 0 \\ H & K & 1 \end{pmatrix}$$

如此點（X，Y）變換結果成：

$$(X, Y, 1)\begin{pmatrix} COS(\theta) & SIN(\theta) & 0 \\ -SIN(\theta) & COS(\theta) & 0 \\ H & K & 1 \end{pmatrix}$$

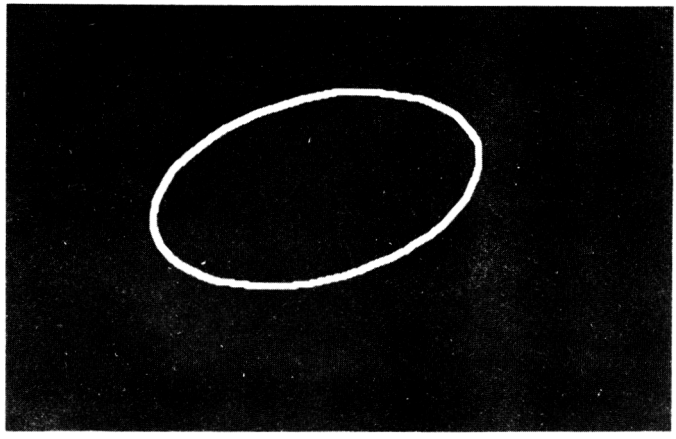
$$= (XCOS(\theta) - YSIN(\theta) + H, XSIN(\theta) + YCOS(\theta) + K, 1)$$

將此變換代入程式5.8產生程式5.9：

程式5·9：橢圓之二

```
1  REM PROGRAM 5.9 (ELLIPSE 2)
2  REM ROTATION; TRANSLATION
10 CX = 140:CY = 96:SC = 1.16:FL = 0:A = 90:
   B = 50
20 H = 10:K = 30:T = .3:C1 = COS (T):
   S1 = SIN (T)
30  HGR2 : HCOLOR= 3
40  FOR TH = 0 TO 6.4 STEP .3
50 X = A * COS (TH):Y = B * SIN (TH
```

```
60 X1 = X * C1 - Y * S1 + H:
   Y1 = X * S1 + Y * C1 + K
70 SX = SC * X1 + CX:SY = CY - Y1
80  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 120
90  IF FL = 1 THEN 110
100  HPLOT SX,SY:FL = 1
110  HPLOT TO SX,SY
120  NEXT TH
```



多型變換（more transformation）

我們可以認爲橢圓只是圓的變形，只要改變圓的X或Y軸方向的比例，就可以改變圓成橢圓。例如，半徑爲1的圓（$X^2 + Y^2 = 1$）我們可以伸展成橢圓

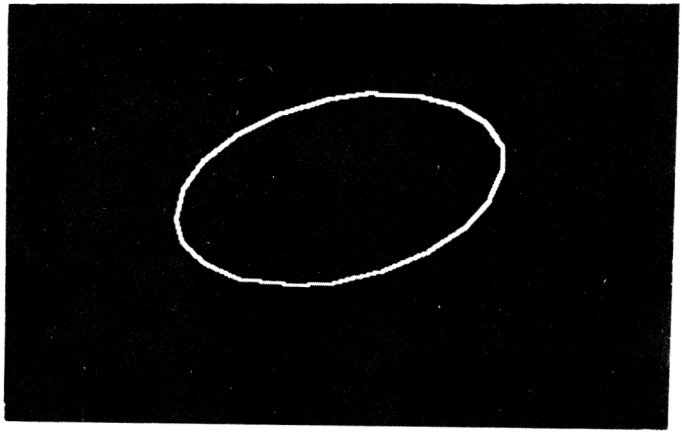
$$\frac{X^2}{A^2} + \frac{Y^2}{B^2} = 1$$

由於X成份有A比例及Y成份B比例（比較5.2式與5.5式），所以比例矩陣可寫成

$$\begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

程式5·10：橢圓之三

```
1   REM PROGRAM 5.10 (ELLIPSE 3)
2   REM ELLIPSE AS A TRANSFORMED CIRCLE
10  CX = 140:CY = 96:SC = 1.16:X = 1:Y = 0:FL = 0
20  A = 90:B = 50
30  DT = .1:C = COS (DT):S = SIN (DT):N = 6.4 /
    DT
40  H = 10:K = 30:T = .3:S1 = SIN (T):C1 = COS
    (T)
50   HGR2 : HCOLOR= 3
60   FOR I = 1 TO N
70  T = X * C - Y * S:Y = Y * C + X * S:X = T
80  X1 = X* A * C1 - Y * B * S1 + H:Y1 = X * A *
    S1 + Y * B * C1 + K
90  SX = SC * X1 + CX:SY = CY - Y1
100   IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
      THEN FL = 0: GOTO 140
110   IF FL = 1 THEN 130
120   HPLOT SX,SY:FL = 1
130   HPLOT TO SX,SY
140   NEXT I
150   END
```



將比例、旋轉、平移的像代入程式5.7後，我們得到另一個畫橢圓程式。

參數型方程式 （Parametric Equations）

如下式的方程式

$$X = T^2 + 10$$

$$Y = T^2 - 10T$$

我們叫做“參數型方程式”，乃利用參數T來求得X及Y值，雖然先前我們未曾提過參數型，但相信讀者已經學過求值的技巧了。因爲程式5.5本身就是參數型方程式的應用

$$X = RCOS(T)$$

$$Y = RSIN(T)$$

當設R = 90 時即可畫圓。先前有些橢圓程式亦是應用此方法來畫橢圓。

因爲參數與X及Y值應用的重要，因此人們常使用參數型方程式把數學應用在科學、工程學及商業的領域。因此想要得到一個平滑的軌跡而減少運算時間，則我們首先得使用參數來定義X與Y間的關係。參數型方程式也可畫一些設計者欲要或奇特的軌跡喔！

利用下式來求得程式5.11之圖案

$$X = 50SIN(2(T-\frac{\pi}{13})$$

$$Y = -70COS(T)$$

以上可改寫爲參數型的一般式：

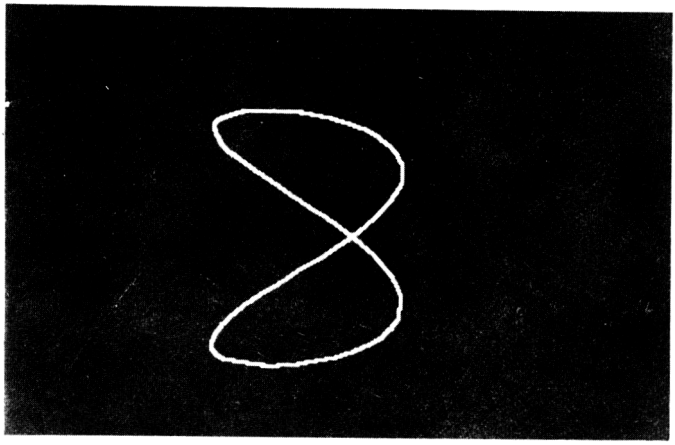
$$X = ASIN(B(T-\frac{\pi}{C}))$$

$$Y = DCOS(T)$$

而定A = 50，B = 2，C = 13，D = -70 繪圖。讀者可自訂A，B，C和D之值，看其結果會如何？

程式5·11：參數型

```
1  REM PROGRAM 5.11 (PARAMETRIC)
2  REM DRAWS LISSAJOUS FIGURES
10 CX = 140:CY = 96:SC = 1.16:FL = 0
20 PI = 3.14159
30 A = 50:B = 2:C = 13:D = -70
40  HGR2 : HCOLOR= 3
50  FOR T = 0 TO 6.3 STEP .1
60 X = A * SIN (B * (T - PI / C)):Y = D * COS
   (T)
70 SX = SC * X + CX:SY = CY - Y
80  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 110
90  IF FL = 0 THEN HPLOT SX,SY:FL = 1
100  HPLOT TO SX,SY
110  NEXT T
```



尚有許多令讀者感興趣的參數型方程式：

$$X = 30T^2$$

$$Y = 30T^3 \quad , -2 \leq T \leq 2$$

$$X = 80\cos T$$

$$Y = 80SINT,\ 0 \leq T \leq 2\pi$$

$$X = 30T - 50SINT$$
$$Y = 30 - COST,\ -4 \leq T \leq 4$$

$$X = 5(COST + TSINT)$$
$$Y = 5(SINT - TCOST),\ 0 \leq T \leq 20$$

除了上面的參數型外，先前例題求得的軌跡不也是參數型的應用。

拋物線（Parabolas）

拋物線的軌跡證明了許多物理學上的應用。譬如，將小石頭或小球向下投擲，則這沿行的路徑（忽略風的阻力）就是一個拋物線的路徑；另實用的“拋物柱面反射鏡（parabolic reflectors）”能集中焦點所得到的反射光於一點上，它也是拋物線軌跡的一種解釋；還有慧星繞着太陽在天體運行的軌道也是拋物線軌跡，而這拋物線的焦點就是太陽。

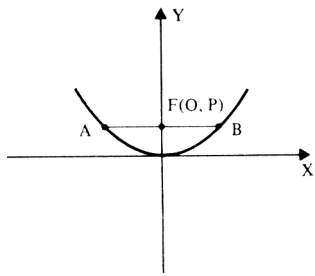


圖 5-9 拋物線

式 $Y = X^2 / 4P$ 的拋物線軌跡如圖 5.9。此拋物線的頂點在原點（0，0）上；焦點在（0，P）。而我們也定拋物線的軸須包含拋物線的焦點及頂點。另 P 值也暗示了頂點和焦點的相對位置及此拋物

線的大略軌跡。至於通過焦點而垂直拋物線軸的AB 線段，我們叫做“正焦弦”其長爲4|P| 。因此由 4 |P|值可判斷出此拋物線的“寬度”若干。

雖然此$Y = X^2 / 4P$可定義出拋物線的軌跡，但是筆者仍樂意將它化成參數型方程式來說明

$$X = 2PT$$

$$Y = PT^2$$

把此數學方程式寫成程式，則我們可以改變P值而很快的求得不同的拋物線。讓讀者改變數種P值於程式5.12 。

若想很輕易的處理一個任意位置及方向的拋物線時——我們必須使用旋轉和平移的方法變換拋物線的點。請讀者試試程式5.13吧！

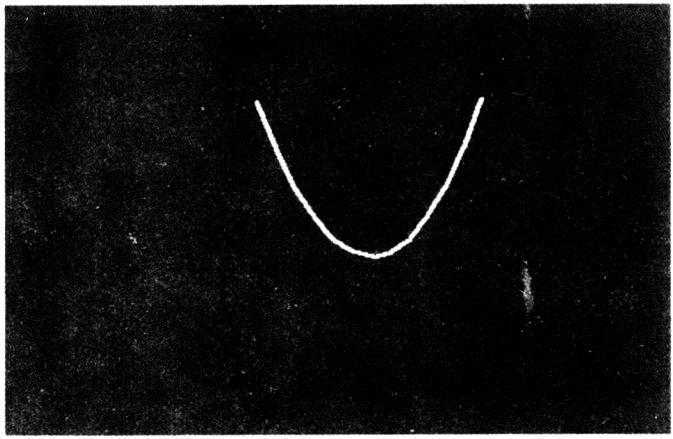
雙曲線（Hyperbolas）

圓、橢圓和拋物線都是屬於圓錐曲線，因此它們軌跡的形成都是由圓錐曲線和平面的交點的總集合。而這雙曲線則是圓錐曲線的另一種軌跡。讓我們一同簡單的來說明另一種新的曲線軌跡——雙曲線。

雙曲線（圖5.10）的數學表示式

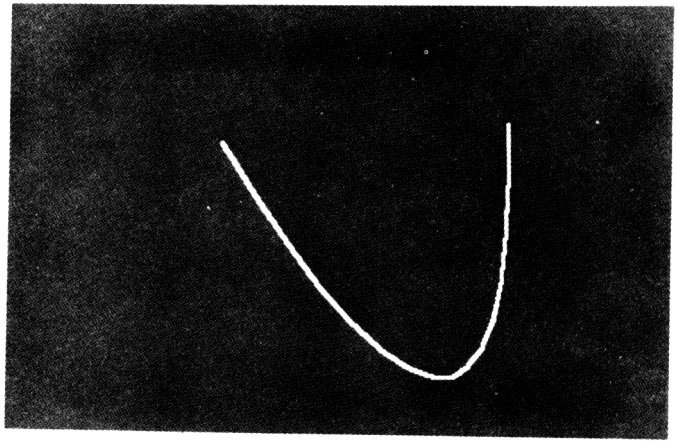
程式5·12：拋物線

```
1   REM PROGRAM 5.12 (PARABOLA)
2   REM DRAWS A PARABOLA; PARAMETRIC EQUATIONS
10  CX = 140:CY = 96:SC = 1.16:FL = 0
20  P = 10
40   HGR2 : HCOLOR= 3
50   FOR T = -5 TO 5 STEP .5
60  X = 2 * P * T:Y = P * T * T
80  SX = SC * X + CX:SY = CY - Y
90   IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
     THEN FL = 0: GOTO 120
100   IF FL = 0 THEN HPLOT SX,SY:FL = 1
110   HPLOT TO SX,SY
120   NEXT T
```



程式 5·13：拋物線的旋轉

```
1  REM PROGRAM 5.13 (ROTATED PARABOLA)
2  REM DRAWS A TRANSLATED, ROTATED PARABOLA
10 CX = 140:CY = 96:SC = 1.16:FL = 0
20 P = 10
30 H = 10:K = -80:TH = .3:S1 = SIN (TH):C1 = COS
   (TH)
40  HGR2 : HCOLOR= 3
50  FOR T = -5 TO 5 STEP .5
60 X = 2 * P * T:Y = P * T * T
70 X1 = X * C1 - Y * S1 + H:Y1 = X * S1 + Y * C1
   + K
80 SX = SC * X1 + CX:SY = CY - Y1
90  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 120
100  IF FL = 0 THEN HPLOT SX,SY:FL = 1
110  HPLOT TO SX,SY
120  NEXT T
```



$$\frac{X^2}{A^2} - \frac{Y^2}{B^2} = 1$$

由此明顯的A和B值的設定影響到雙曲線的形狀。其參數表示式

$$X = A\,SECT$$

$$Y = B\,TANT$$

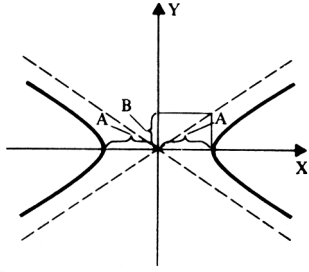
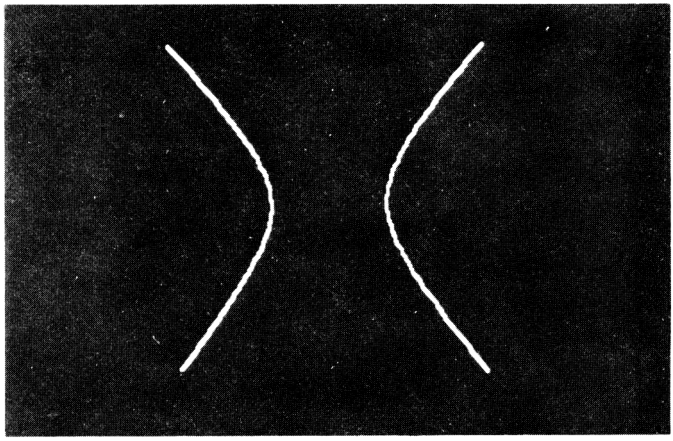


圖 5-10　雙曲線

它也表示了雙曲線，我們就利用此參數寫程式而繪圖。

程式 5·14：雙曲線

```
1  REM PROGRAM 5.14 (HYPERBOLA)
2  REM PARAMETRIC EQUATIONS FOR A HYPERBOLA
10 CX = 140:CY = 96:SC = 1.16:FL = 0
20 A = 30:B = 35
30  HGR2 : HCOLOR= 3
40  FOR T = -1.6 TO 4.5 STEP .05
50 X = A / COS (T):Y = B * TAN (T)
60 SX = SC * X + CX:SY = CY - Y
70  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 100
80  IF FL = 0 THEN HPLOT SX,SY:FL = 1
90  HPLOT TO SX,SY
100  NEXT T
```



極座標曲線（Polar Coordinate Curves）

程式5.5應用了參數方程式來繪圓

$$X = R\,COS(T)$$

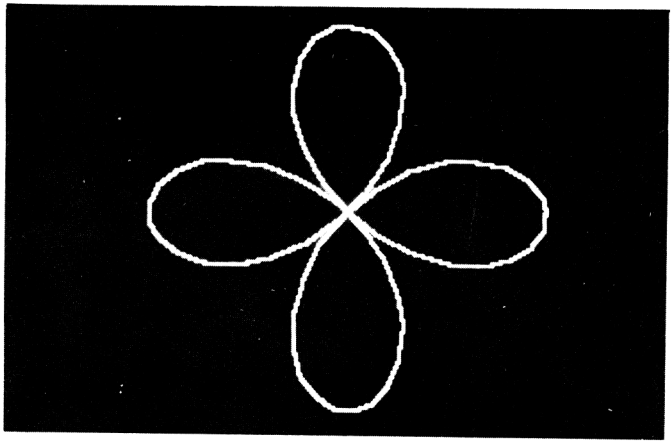
$$Y = R\,SIN(T)$$

此圓的半徑R = 90 。

如果讀者致使R變化，以代替先前的定值，則我們得到一個非常有趣及令人興奮的軌跡。因此此節我們設定R當作T的函數，因此聯想到T的改變導致R函數的變化。

程式5·15：極點

```
1   REM PROGRAM 5.15 (POLAR)
2   REM DRAWS A ROSE
10 CX = 140:CY = 96:SC = 1.16:FL = 0
20   HGR2 : HCOLOR= 3
30   FOR T = 0 TO 6.3 STEP .1
40 R = 80 * COS (2 * T)
50 X = R * COS (T):Y = R * SIN (T)
60 SX = SC * X + CX:SY = CY - Y
70   IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
     THEN FL = 0: GOTO 100
80   IF FL = 0 THEN HPLOT SX,SY:FL = 1
90   HPLOT TO SX,SY
100   NEXT T
```



首先向讀者說明的，事實上R和T之間的關係實在的影響了圖形軌跡的變化。讀者不妨做以下的實驗來證明，試

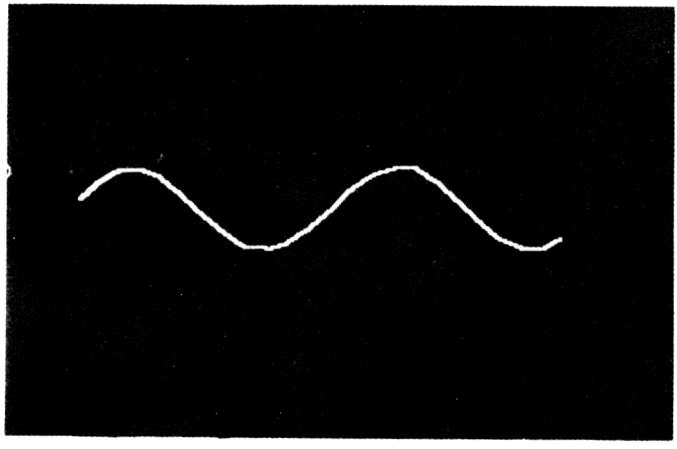
```
R = 80*COS(3*T) (Try 0 ≤ T ≤ 3.2 STEP .1)
R = 80*COS(4*T) (Try 0 ≤ T ≤ 6.3 STEP .1)
R = 30*(COS(T) - 1)
R = 30*(2*COS(T) - 1)
R = 2*T (Try 0 ≤ T ≤ 70 STEP .2)
```

函數型（Form Y=F(X)）

在此值得一提的函數型Y＝F（X）的軌跡可利用先前幾節的討論來求得點（X，Y）座標而繪出。程式 5.16 說明方法之一，行號

程式 5·16：函數軌跡

```
1  REM PROGRAM 5.16 (CURVE)
2  REM DRAWS A SINE CURVE
10 XS = 20:YS = 20:MINX = -6:MAXX = 6:ST = .5
20 FL = 0:CX = 140:CY = 96:SC = 1.16
30  DEF FN Y(X) = SIN (X)
40  HGR2 : HCOLOR= 3
50  FOR X = MINX TO MAXX STEP ST
60 Y = FN Y(X)
70 SX = CX + SC * XS * X:SY = CY - YS * Y
80  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
    THEN FL = 0: GOTO 110
90  IF FL = 0 THEN HPLOT SX,SY:FL = 1
100  HPLOT TO SX,SY
110  NEXT X
```



30的典型函數的變化是隨著其範圍（MINX＜X＜MAXX）、步增值（ST）和比例因素（XS，YS）之改變而改變。所以此程式必須詳盡地先設定範圍、步增值和比例因素方可決定適當軸和允許使用者進行此函數。

第四部份

立體圖形

第六章　立體圖形的基本原理

電腦的顯示幕只是一個平面的顯示裝置，然而就如同藝術家能在畫布上創造出一個立體景物的印象一樣，我們也能在繪圖電腦上顯示立體圖形。

> 注意——仔細研讀應用的數學
>
> 在做顯示立體圖形的處理時，必須牽涉到一些數學理論，讀者須仔細研讀。其道理就和在崎嶇不平的地形上行走時一樣，若我們小心，謹愼的前進，將會很快的走完行程。而欲速則不達。

爲了要產生立體圖形，我們必須訂定一些法則來確定立體空間中的點。底下我們將討論兩種座標系統。

座標系統 （Coordinate Systems）

在直角座標系統中需要三個互相垂直的座標軸，除X和Y軸構成一平面系統外，另外再增加一經過原點並垂直XY平面的Z軸。對Z軸而言，其方位測定有兩種。圖6.1所示爲右手系統，利用右手的食指，中指和姆指各代表X，Y和Z軸的方向。

而左手系統唯一不同之處卽其Z軸方向相反而已。

若在右手系統中觀察XY平面，則Z軸將垂直這個平面，並朝向觀測者的眼睛，如圖6.2a。而左手系統的Z軸則離開觀測點，如圖6.2b。

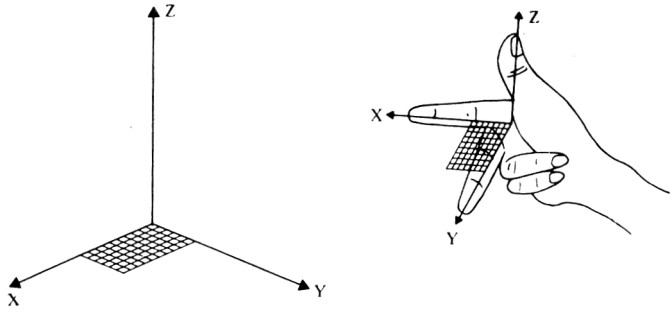


圖 6-1 直角座標系統

至於選擇右手或左手系統並不一定，因爲任一系統均能完成我們想要的計算。但爲方便起見，通常我們都以右手系統爲標準。

任兩個座標軸將形成一平面，稱之爲座標平面。因此將產生三個座標平面：XY平面，YZ平面和XZ平面。在空間中任一點可表示爲（A，B，C），而A，B，C的值也就是點到上述三個平面的個別距離。

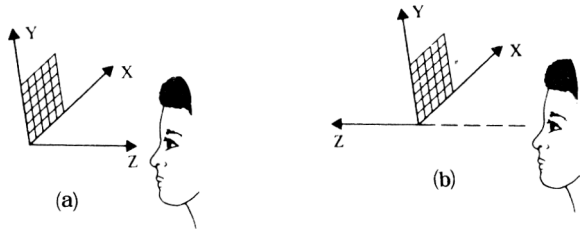


圖 6-2 右手座標系統(a)和左手座標系統(b)

圖6.3中畫出了（1，8，5），（5，3，3），（0，5，−7）和（−2，5，−7）這幾個點。

我們要使用的第二種座標系統爲球體座標系統（如圖6.4）。在這個系統，每一點P可表示爲（ρ，θ，ϕ）。ρ爲點到原點的距離，ϕ爲正向Z軸和OP之間的角度，θ則爲正向X軸和OP在XY平面

的投影之間的角度，並且是從逆時針方向（若從正向Z軸看進去）所測得的。

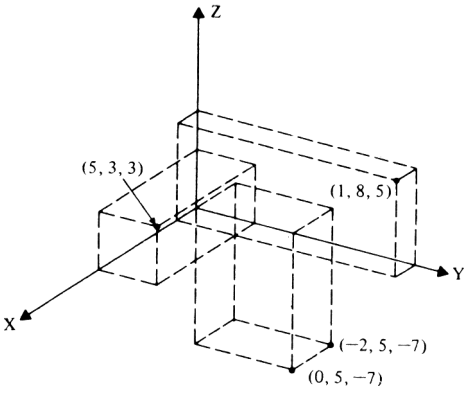


圖 6-3

希臘字母：在一般的數學書籍，對於球體座標常用到一些希臘字母 ρ（rho），θ（theta），和 ϕ（phi），但在BASIC我們用RHO，THETA，和PHI來表示，且因Applesoft BASIC對變數名稱只承認前面兩個字母，所以我們就用RH，TH和PH來表示。在程式6.1中即用到此。

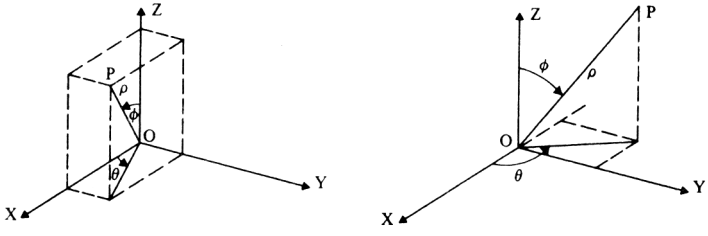


圖 6-4　球體座標系統

因爲一個點可以表示爲直角座標（X，Y，Z）或球體座標（ρ，θ，ϕ），因此兩個座標系統之間必須有一種轉換關係。如下：

$$x = \rho SIN\phi COS\theta$$
$$y = \rho SIN\phi SIN\theta$$
$$z = \rho COS\phi$$
$$\rho^2 = x^2 + y^2 + z^2$$

爲了在平面的圖形顯示幕上得到一個立體圖形。因此我們必須結合顯示幕座標（ SX ， SY）和直角座標（ X ， Y ， Z ）或球體座標（ ρ ， θ ， ϕ ）。圖6.5中卽在獲得一顯示幕座標。

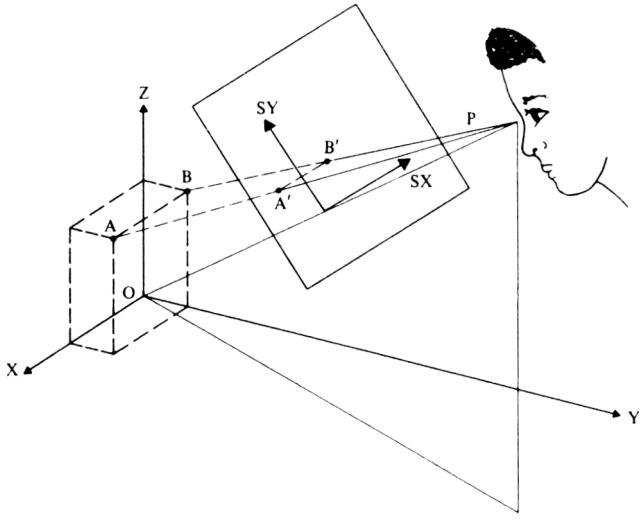


圖 6-5 顯示幕座標的產生

圖中，P點是觀測者的位置，顯示幕平面上是物體的投影。這個投影平面是假設垂直OP 線段，且與P點的距離爲D 。當物體被投影在投影平面上，對每一點（ X ， Y ， Z ）我們可以得到顯示幕座標（ SX ， SY ）如圖6.7 。

如果我們採球體座標來表示觀測點（ 觀測者眼睛的位置 ）的話，那麼觀測點的將很容易被控制：

• ρ（ rho ）表示觀測點到原點（ 0 ， 0 ， 0 ）的距離。若增加ρ將

會使觀測者遠離原先觀測的位置。而造成物體被顯示成一較小的圖形。

- θ（ theta ）和ϕ（ phi ）則決定觀測者要看物體的那一個方向。

在圖6.5中我們安排投影平面（顯示幕）的位置介於觀測者和物體之間。但這並不是一定的，圖6.6中就把顯示幕置於與觀測者相對的一邊。如此將會造成一較大的顯示圖形。即：

- 若使顯示幕較接近觀測者的話，將形成一個較小的投影圖形。
- 若使顯示幕較遠離觀測者的話，將形成一個較大的投影圖形。

控制觀測的助變數ρ，θ，ϕ和D將能控制觀測的位置和方向以及圖形的大小。以後我們還會再提到有關的情形，首先我們必須獲得一些處理立體物體的工具。

變換（Transformations）

在此我們將再次使用均質坐標（ homogenenous coordinates）來表示點。一個點的直角座標是（ X，Y，Z ），若用均質座標則爲（ X，Y，Z，1 ），經由線性變換（ Linear transformations）

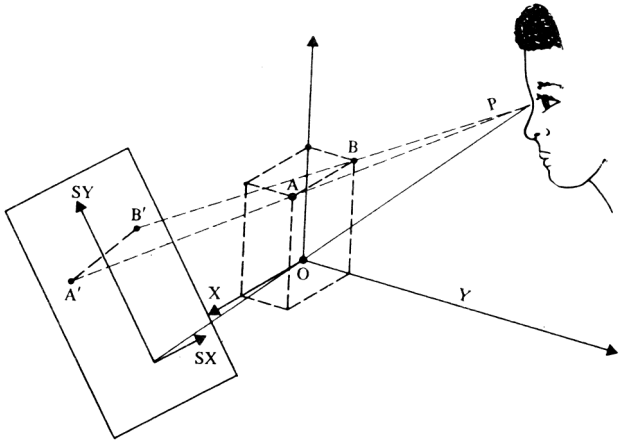


圖6-6

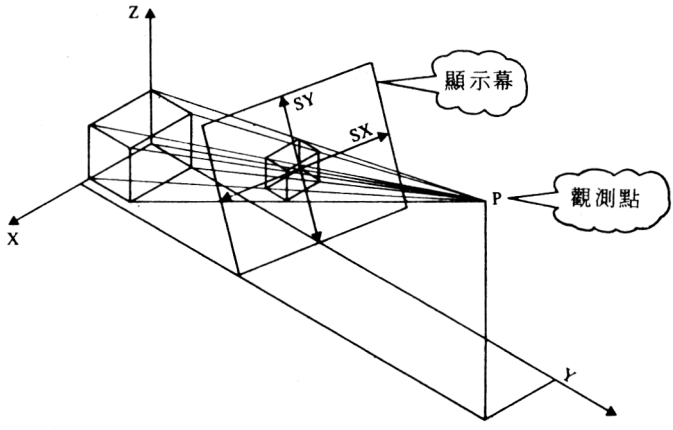


圖 6-7

可知其可以用一個 4 × 4 的矩陣來表示。在此我們並不考慮所有的變換方法，而把焦點放在最重要的幾個。

比例（scaling）

矩陣主對角線上的元素代表比例變換。因爲

$$(X,Y,Z,1)\begin{bmatrix}A&0&0&0\\0&B&0&0\\0&0&C&0\\0&0&0&1\end{bmatrix}=(AX,BY,CZ,1)$$

因此，我們可以個別控制每一座標在任一座標軸方向的比例。若要一致的比例則用：

$$(X,Y,Z,1)\begin{bmatrix}K&0&0&0\\0&K&0&0\\0&0&K&0\\0&0&0&1\end{bmatrix}=(KX,KY,KZ,1)$$

由於（X，Y，Z，D），化成標準式時應爲（X/D,Y/D,Z/D,1）

，因此一致的比例也可經由乘上如下這個矩陣來達成。

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/K \end{bmatrix}$$

旋轉（rotation）

要使物體對應每一個座標軸而旋轉，這是很容易達成的。在第二章，當考慮對原點旋轉時，我們使用下面這個變換矩陣：

$$R = \begin{bmatrix} COS\theta & SIN\theta & 0 \\ -SIN\theta & COS\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

若將一向量（X，Y，1）乘上R，則會使這個向量對原點逆時針方向旋轉一角度θ。假設在XY平面上加上一個朝向我們的Z軸，則其動作就如同繞著Z軸旋轉。若應用在三維（three-dimensional）向量，則應使用這個矩陣：

$$R_z = \begin{bmatrix} COS\theta & SIN\theta & 0 & 0 \\ -SIN\theta & COS\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

若要對X軸和Y軸旋轉時，其變換矩陣如下：

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & COS\theta & SIN\theta & 0 \\ 0 & -SIN\theta & COS\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} COS\theta & 0 & -SIN\theta & 0 \\ 0 & 1 & 0 & 0 \\ SIN\theta & 0 & COS\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

任何一種型式，都是從正向軸看進去逆時鐘旋轉一個角度 θ 。

位移（translation）

如果我們不需要位移的功能時，可以不使用均質座標。利用下面這個矩陣，將可以很容易達成位移的功能。

$$(X,Y,Z,1)\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\H&K&L&1\end{bmatrix}=(X+H,Y+K,Z+L,1)$$

要把一點（或聚集的點）移動一個向量（H，K，L）的距離時，我們可以乘以上面這個矩陣。而結果有時也可以看成是把新原點移到點（－H，－K，－L）的地方。

對稱（reflection）

物體對一平面的對稱，也就是在平面的另外一邊產生如鏡中映像的物體。因此，點對 X Y 平面對稱的結果，將會改變每一點的 Z 軸符號。對稱的變換矩陣爲

$$M_{xy}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$$

若是對 X Z 平面和 Y Z 平面對稱，其變換矩陣分別爲

$$M_{xz}=\begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$M_{yz} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

連續的變換

連續變換的結果可以只用一個矩陣來表示，即個別的變換矩陣之積。例如我們要使射線 PQ（ P =（0，2，3），Q =（1，2，3））旋轉 30°，則

1. 位移

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{bmatrix}$$ 然後再

2. 旋轉

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{COS}30° & \text{SIN}30° & 0 \\ 0 & -\text{SIN}30° & \text{COS}30° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ 然後再

3. 位移

$$T^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 3 & 1 \end{bmatrix}$$

亦即：

$M = TRT^* =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{COS}30° & \text{SIN}30° & 0 \\ 0 & -\text{SIN}30° & \text{COS}30° & 0 \\ 0 & -2\text{COS}30°+3\text{SIN}30°+2 & -2\text{SIN}30°-3\text{COS}30°+3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & .866 & .5 & 0 \\ 0 & -.5 & .866 & 0 \\ 0 & 1.768 & -.598 & 1 \end{bmatrix}$$

在連續變換中，順序是很重要的，若順序改變，其結果亦隨之改變。例如，先旋轉再位移的結果就和先位移再旋轉的結果不同。

逆矩陣（inverses of martrices）

在上個例子中，T和T* 的效果剛好相反。若乘矩陣T將造成位移（0，－2，－3），而乘T* 則造成位移（0，2，3）。同時乘T和T* 則使點回到原來的位置，因爲

$$TT^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

凡兩個矩陣之乘積爲I者卽爲反逆（inverses）。矩陣M的逆矩陣通常表示爲M^{-1}。但並一定每個矩陣都有逆矩陣，逆矩陣對電腦繪圖很有用，例如這個矩陣

$$R=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \mathrm{COS}30^\circ & \mathrm{SIN}30^\circ & 0\\ 0 & -\mathrm{SIN}30^\circ & \mathrm{COS}30^\circ & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .866 & .5 & 0\\ 0 & -.5 & .866 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

表示對X軸旋轉30°，而它的逆矩陣R^{-1}則表示對X軸旋轉－30°：

$$R^{-1}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .866 & -.5 & 0\\ 0 & .5 & .866 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

變換座標系統

我們已經提過在一個座標系統裏變換一個點（或聚集的點），另外我們也經常藉軸的變換而建立一個新的座標系統。例如我們可以把原點移到一個新的位置，或旋轉座標系統。

如此，我們就必須把一個點以原來（或標準）的座標以及新座標系統的座標表示。而對於第二個軸系統的座標我們加上一個符號（′）。因此座標（2，1，3）爲標準的軸系統，而（2，1，3）′則表第二個軸系統。

例如，對Z軸逆時針旋轉30°，其變換矩陣爲：

$$M=\begin{bmatrix}\mathrm{COS}30^\circ & \mathrm{SIN}30^\circ & 0 & 0\\ -\mathrm{SIN}30^\circ & \mathrm{COS}30^\circ & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$= \begin{bmatrix} .866 & .5 & 0 & 0 \\ -.5 & .866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

若把這個變換矩陣乘上一個點座標向量，將會得到旋轉點的座標向量。

例如點（2,4,1）將旋轉到（－.268，4.464，1）的位置，因爲：

$$(2,4,1,1) \begin{bmatrix} .866 & .5 & 0 & 0 \\ -.5 & .866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (-.288, 4.464, 1, 1)$$

從P點對Z軸逆時針旋轉30°將得到P′點。然後我們看座標系統也對Z軸順時針旋轉30°後的現象。新的座標系統X′，Y′，Z′，對P點有另外的意義：P′點對應在X，Y，Z座標系統的位置和P點對應在X′，Y′，Z′座標系統的位置一樣。

因此，矩陣M可作爲：

1. 決定一個點在經過逆時針方向旋轉30°後的座標。或是
2. 決定一個在經過順時針方向旋轉30°後的座標上之點座標。

上面這個例子的型式並不是一定的，它只是用來說明一個重要的原理：用來變換軸系統的矩陣亦即軸系統裏變換點的矩陣之逆矩陣。

投影（projection）

現在我們已經知道了如何處理立體圖形，但我們還不準備把它顯示在電腦的顯示幕。當然，理由是顯示幕只是一個二維（two-dimensional）平面。爲了把一個立體圖形顯示在平面的顯示幕上，底下我們發展出一種方法把物體的每一點投影在一個平面上，如圖6.8所示。在面對這個問題之前，我們將考慮一個有關聯的問題：點

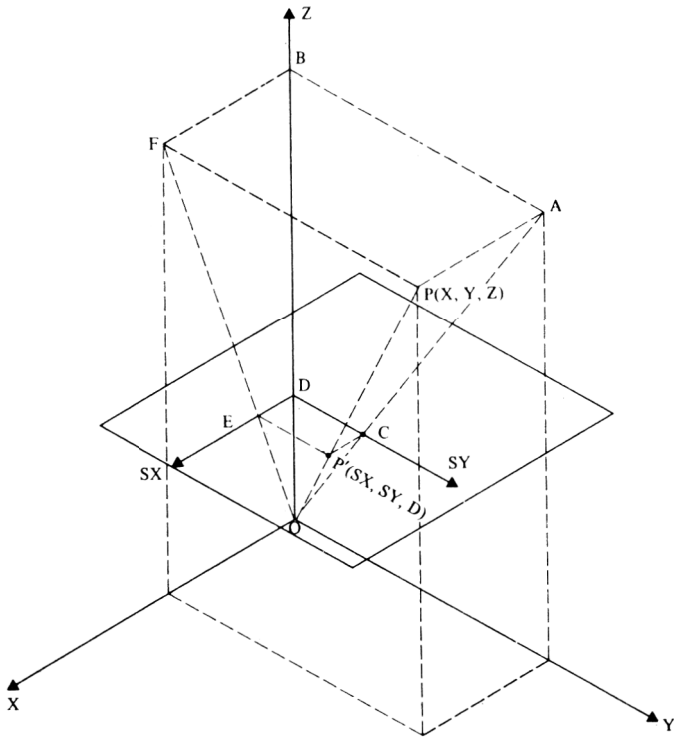


圖 6-8

在一平行於 XY 平面的平面上之投影。

投影平面位於距XY平面D的位置，如圖 6.8 所示。而投影平面也就將是顯示幕，所以我們定義此平面上的點爲顯示幕座標（ SX，SY ），且 SX 軸和 SY 軸各平行於 X 和 Y 軸。在投影平面上所量出距離的比例將和座標軸一樣。故平面上一點（ SX，SY ）被引據自（ X ，Y ，D ），且 X = SX ，Y = SY 。

投影的方法有很多種，在此我們將使用透視投影（ perspective projections ），取出投影到原點O（ 0 ，0 ，0 ）的中央，也就是一點 P（ X ，Y ，Z ）將會被投影在投影平面和線段 OP 的交點。

爲了要確定顯示幕座標（SX,SY）係結合自P點（X，Y，Z），注意直角三角形OBA和ODC（圖6.8和圖6.9所示）是爲相似三角形，故

$$\frac{DC}{OD}=\frac{BA}{OB}$$

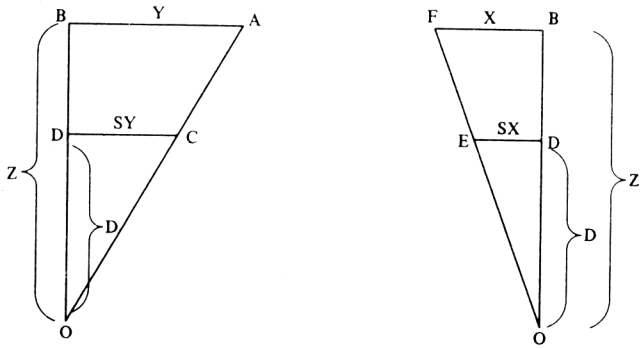


圖 6-9 相似三角形

因此

$$\frac{SY}{D}=\frac{Y}{Z}$$

所以SY = D（Y / Z）。

同理，從相似三角形OBF和ODE我們可以得到

$$\frac{DE}{OD}=\frac{BF}{OB} \quad 和 \quad \frac{SX}{D}=\frac{X}{Z}$$

因此SX = D（X / Z）。

現在我們總算具備了在顯示幕顯示立體圖形的必備條件，讓我們開始進行吧！

立體物體的平面圖形 （Two-Dimensional Images

Of Three-Dimensional Objects）

在圖6.5中，眼睛的位置（P點）定義爲直角座標（H，K，L）和球體座標（ρ，θ，ϕ）。而投影平面（即顯示幕）放在距離觀測者眼睛D的位置，並垂直於觀測者眼睛和原點間的直線。P點就形成了第二個直角座標系統。因此點的座標就可定義爲眼睛座標（Xe，Ye，Ze）或標準座標（X，Y，Z）。

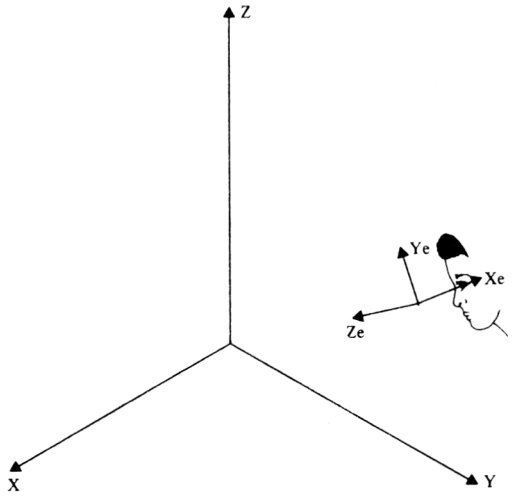


圖 6-10　眼睛座標系統

對於眼睛座標系統如圖6.10所示，當觀測者的視線正對著X，Y，Z軸系統時，則Ze軸就是在這條視線上，而Xe軸朝右邊，Ye軸朝上面。（注意，此時眼睛座標系統即爲左手系統。）

現在我們的任務就分爲兩個階段：

1. 把標準座標（X，Y，Z）變換成眼睛座標（Xe，Ye，Ze）。
2. 把眼睛座標（Xe，Ye，Ze）變換成顯示幕座標（SX，SY）。

第二個階段在前面我們已經計算過了，從上一節我們得到等式SX＝D（Xe／Ze）和SY＝D（Ye／Ze）。

現在我們就進行第一個階段，建立標準座標和眼睛座標的關係，

這可以分爲四個步驟，一步一步的把X，Y，Z的軸系統變成Xe，Ye，Ze的軸系統，每一步的結果我們用插圖表示在圖6.11至圖6.14，每一個變換後的軸系統都用X′，Y′，Z′表示，而最後一個X′，Y′，Z′軸系統也就是眼睛座標系統Xe，Ye，Ze（比較圖6.14和圖6.5）。

當建立新座標系統的程序完成之後，我們就得到了眼睛座標系統。

這些轉換的目的就在於得到P點（觀測者的眼睛）在標準座標系統上的直角座標(H, K, L)或球體座標（ρ，θ，ϕ）。且$H=\rho\sin\phi\cos\theta$，$K=\rho\sin\phi\sin\theta$，$L=\rho\cos\phi$。

簡要的說，即

1. 第一步在觀測者的眼睛建立一個座標系統，和原來軸系統的方位和方向都一樣，這個處理可以由下面將提到的矩陣A完成。
2. 第二步和第三步則旋轉軸系統使Z′軸朝向標準系統的原點，而Y′軸的直線方向和Z軸相交。若從觀測者的位置來看，則Y′軸朝上，X′軸朝左。
3. 第四步把X′軸翻軸形成一左手系統，以提供一個比較便利的方位給觀測者。

因爲變換過程最好的表示方法就是矩陣，因此我們把每一個步驟建立一個矩陣。

第一個步驟建立一個新原點（H，K，L）=（ρSINϕCOSθ，ρSINϕSINθ，ρCOSϕ）。其變換矩陣如下：

$$A=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -H & -K & -L & 1 \end{bmatrix}$$

或

$$A=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\rho SIN\phi COS\theta & -\rho SIN\phi SIN\theta & -\rho COS\phi & 1 \end{bmatrix}$$

雖然球體座標比較不好表示，但在計算時卻比較方便。

第二步對Z′軸順時針旋轉一個角度（$90°-\theta$）。而旋轉之後的結果，負向Y′軸的直線方向和Z軸相交。其變換矩陣如下：

$$B=\begin{bmatrix} SIN\theta & COS\theta & 0 & 0 \\ -COS\theta & SIN\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

（記住：用來變換軸系統的矩陣即爲軸系統裏變換點的矩陣之逆矩陣。）

第三步把軸系統對X′軸逆時針旋軸（$180°-\phi$）。而使Z′軸朝向X，Y，Z軸系統的原點。其矩陣爲

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -COS\phi & -SIN\phi & 0 \\ 0 & SIN\phi & -COS\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

第四步，翻轉X′軸，使之變成左手座標系統。

$$D=\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

若已知一點之標準座標（X，Y，Z）則其眼睛座標（Xe，Ye，Ze）即可由（Xe，Ye，Ze）=（X．Y，Z）ABCD 得到。爲便利計算，我們用一個矩陣代替，T = ABCD。因此

$$(Xe, Ye, Ze, 1) =$$

$$(X, Y, Z, 1)\begin{bmatrix} -SIN\theta & -COS\theta COS\phi & -COS\theta SIN\phi & 0 \\ COS\theta & -SIN\theta COS\phi & SIN\theta SIN\phi & 0 \\ 0 & SIN\phi & -COS\phi & 0 \\ 0 & 0 & \rho & 1 \end{bmatrix}$$

至於 ABCD 相乘的結果則爲

$$T = ABCD$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\rho\cos\theta\sin\phi & -\rho\sin\theta\sin\phi & -\rho\cos\phi & 1 \end{bmatrix}$$

$$\begin{bmatrix} \sin\theta & \cos\theta & 0 & 0 \\ -\cos\theta & \sin\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & -\cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sin\theta & \cos\theta & 0 & 0 \\ -\cos\theta & \sin\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\rho\sin\phi & -\rho\cos\phi & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & -\cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sin\theta & -\cos\theta\cos\phi & -\cos\theta\sin\phi & 0 \\ -\cos\theta & -\sin\theta\cos\phi & -\sin\theta\sin\phi & 0 \\ 0 & \sin\phi & -\cos\phi & 0 \\ 0 & 0 & \rho & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\sin\theta & -\cos\theta\cos\phi & -\cos\theta\sin\phi & 0 \\ \cos\theta & -\sin\theta\cos\phi & -\sin\theta\sin\phi & 0 \\ 0 & \sin\phi & -\cos\phi & 0 \\ 0 & 0 & \rho & 1 \end{bmatrix}$$

一、原點移到（H，K，L）：

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\rho\mathrm{COS}\theta\mathrm{SIN}\phi & -\rho\mathrm{SIN}\theta\mathrm{SIN}\phi & -\rho\mathrm{COS}\phi & 1 \end{bmatrix}$$

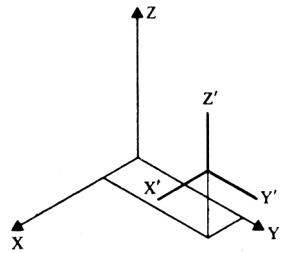


圖 6-11

二、對Z′軸順時針旋轉（$90° - \theta$）

$$B = \begin{bmatrix} SIN\theta & COS\theta & 0 & 0 \\ -COS\theta & SIN\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

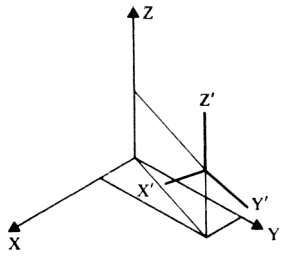


圖 6-12

三、對 X′ 軸逆時針旋轉（ $180° - \phi$ ）：

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -COS\phi & -SIN\phi & 0 \\ 0 & SIN\phi & -COS\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

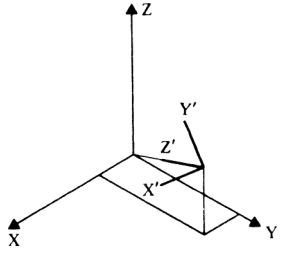


圖 6-13

四、變成左手系統：

$$D = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

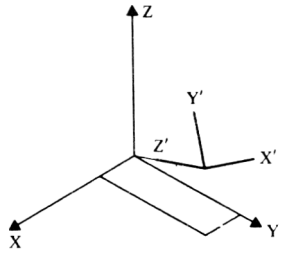


圖 6-14

程式設計

矩陣已經提供給我們一個很便利的型式來建立標準和眼睛座標之間的關係，但微電腦可用的語言中，通常不能提供矩陣的計算。對於上面這些矩陣的積，在BASIC程式中我們這樣寫：

```
10 S1 = SIN(THETA): C1 = COS(THETA):
   S2 = SIN(PHI): C2 = COS(PHI)
20 XE = -X*S1 + Y*C1
30 YE = -X*C1*C2 - Y*S1*C2 + Z*S2
40 ZE = -X*C1*S2 - Y*S1*S2 - Z*C2 + RHO
```

然後顯示幕座標就可以這樣寫

```
50 SX = D*XE/ZE
60 SY = D*YE/ZE
```

不同的微電腦在使用顯示幕定址技術（screen addressing techniques）時，都需要把行號50和60做一個修正，若在APPLE Ⅱ顯示幕要把圖形顯示在中央，則

```
50 SX = D*XE/ZE + 140
```

```
60 SY = 96 - D*YE/ZE
```

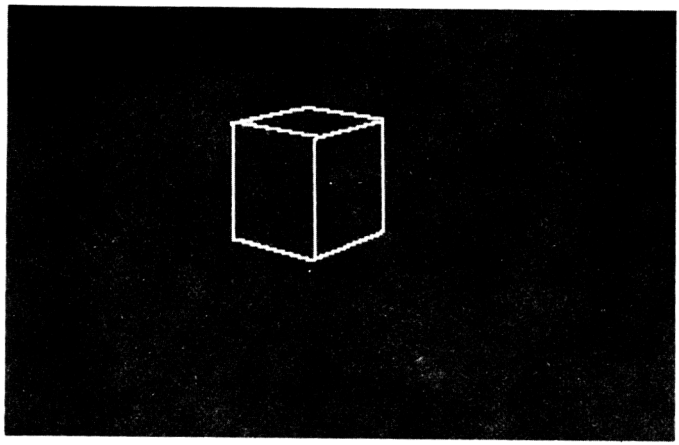
程式6.1提供一種方法來使用上面的公式產生立體物體。行號10定義觀測的助變數。注意我們是選擇D比RHO大，而使物體介於觀測者和投影平面之間。行號20計算在投影計算中須用到的值。行號30定義高解像度圖形顯示幕的中央。行號40到60則提供一個立方體頂點的座標。行號80到180則把這些座標從DATA陳述語讀出來，並提供給行號150到160計算投影顯示幕座標SX和SY的副程式。從副程式回來後，就把投影的點畫上去。

觀測點和透視法（viewpoint and perspective）

RHO，THETA，PEI和D對顯示幕圖形的影響，倒很值得我們注意。

程式6·1：立方體（CUBE）

```
1   REM PROGRAM 6.1  (CUBE)
2   REM DRAWS THE VISIBLE PARTS OF A CUBE
10  RHO = 10:THETA = .7:PHI = 1.3:D = 500
20  S1 = SIN (TH):C1 = COS (TH):S2 = SIN (PH):
    C2 = COS (PH)
30  CX = 140:CY = 96
40   DATA 1,1,0, 1,1,1, 0,1,1, 0,1,0
50   DATA 1,1,0, 1,0,0, 1,0,1, 0,0,1
60   DATA 0,1,1, 1,0,1, 1,1,1
70   HGR2 : HCOLOR= 3
80   READ X,Y,Z: GOSUB 150: HPLOT SX,SY
90   FOR I = 1 TO 8: READ X,Y,Z
100   GOSUB 150: HPLOT  TO SX,SY
110   NEXT I
120   READ X,Y,Z: GOSUB 150: HPLOT SX,SY
130   READ X,Y,Z: GOSUB 150: HPLOT TO SX.SY
140   END
150 XE = - X * S1 + Y * C1:YE = - X * C1 *
    C2 - Y * S1 * C2 + Z * S2:ZE = - X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
160 SX = D * XE / ZE + CX:SY = CY - D * YE / ZE:
    RETURN
```



改變THETA和PHI的值將允許我們從不同的角度觀測物體，改變RHO 的值（觀測者到原點的距離）則允許觀測者靠近或遠離物體，即控制圖形的大小。

另外一種控制圖大小的方法是改變D，若觀測點（ρ，θ，ϕ）不變，則改變D將影響圖形在投影平面上的大小——D增加時，圖形將變大。圖6.15中顯示RHO和D變化所產生的結果。

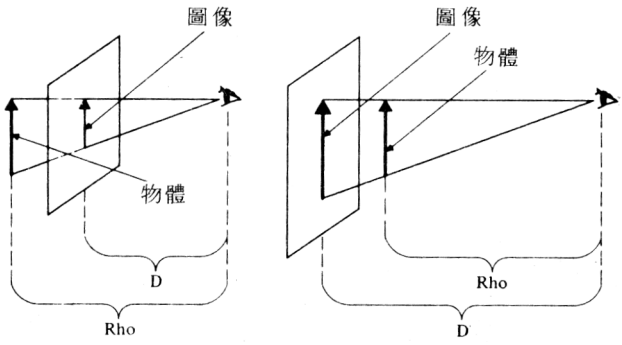


圖6-15　RHO和D對顯示圖形的影響

因爲 RHO 和 D對圖形的大小都有影響，所以了解它們之間的關係是很重要的。如果它們的作用都是控制大小，爲什麼要用兩個控制？原因就在於透視法。一個物體，較靠近觀測者的部分會顯得比較遠的大，因爲它在視野佔了較大的比例。因此，透視觀測一個立方體將造成較近的一邊比較遠的一邊大。若要減少透視對外觀的影響，則增加RHO 的值，但這樣會使圖形看起來變小，所以要增加D來補償。（如果RHO / D的比值一定時又將如何？）

程式 6.2可用來觀察RHO，THETA，PHI 和D的影響。程式在接受 RHO，THETA，PHI和D的值之後，就用這些值顯示出觀察到的立方體。爲便於維持方位起見，在立方體面對正向X軸的一面，我們加上一條對角線。試試看下面這些值的組合，並輸入程式中試驗。（注意，THETA和PHI 的值是弧度）

RHO	7	3	15	15	15	15
THETA	.3	.3	.3	.7	1.5	2.2
PHI	1.3	1.3	1.0	1.3	1.3	1.3
D	150	60	300	300	300	300

程式 6.2說明了RHO，THETA，PHI 和D這些值的功用。並也指出了另一種關係：我們可以“看透”這個立方體。這在某些場合是可以接受的，且某些場合是想要的。但也形成一個問題，就是它帶來了視覺上的錯覺。下一章我們將討論兩個消去法（eliminating）來“隱藏”顯示幕的線和面。

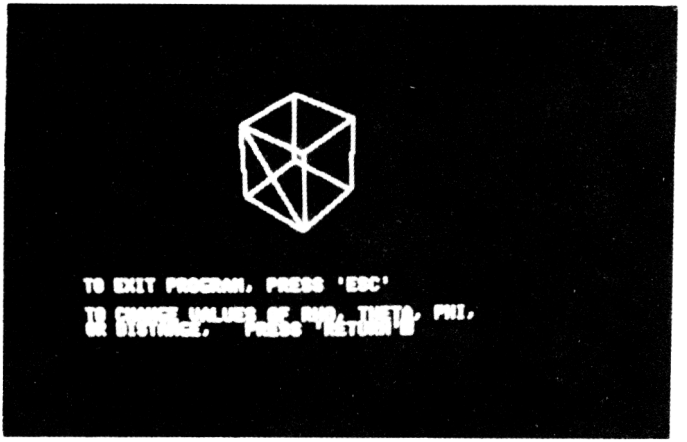
程式 6·2：觀測點

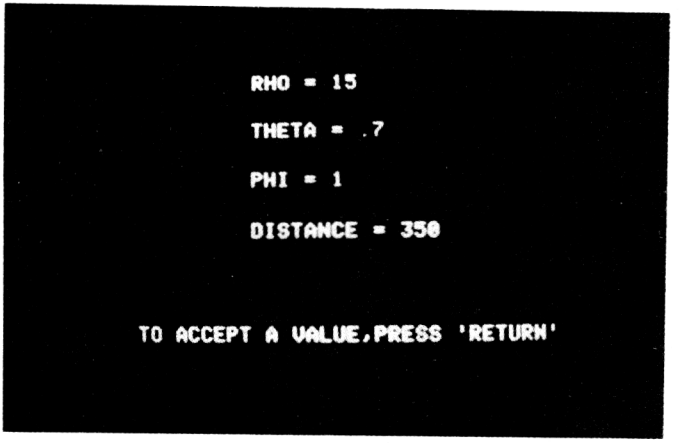
```
1   REM PROGRAM 6.2 (VIEWPOINT)
2   REM ILLUSTRATES EFFECT OF CHANGING THE
    VIEWPOINT PARAMETERS
10   TEXT : HOME : GOTO 180
20 XE = - X * S1 + Y * C1:YE = - X * C1 *
    C2 - Y * S1 * C2 + Z * S2:ZE = - X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
```

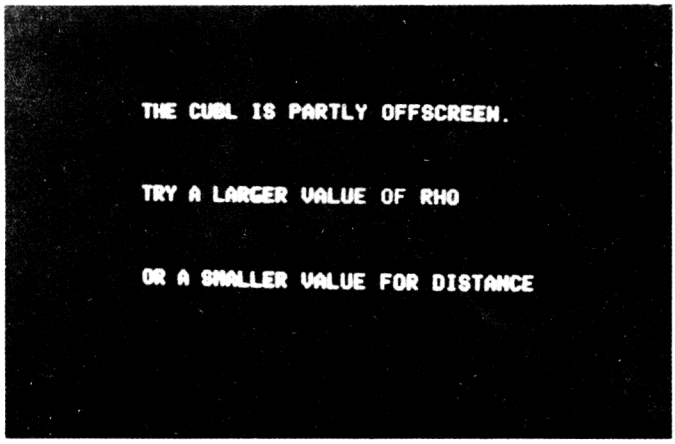
(continued)

```
30 SX = D * XE / ZE + CX:SY = CY - D * YE / ZE:
   RETURN
40  HGR : ONERR GOTO 430
50 S1 = SIN (THETA):C1 = COS (THETA)
60 S2 = SIN (PHI):C2 = COS (PHI)
70  RESTORE : READ X,Y,Z: GOSUB 20: HPLOT SX,SY
80  FOR I = 1 TO 11: READ X,Y,Z: GOSUB 20:
    HPLOT TO SX,SY: NEXT I
90  FOR I = 1 TO 2: READ X,Y,Z: GOSUB 20:
    HPLOT SX, SY
100  READ X,Y,Z: GOSUB 20: HPLOT TO SX,SY:
     NEXT I
110  VTAB 21: PRINT "TO EXIT PROGRAM,
     PRESS 'ESC'"
120  VTAB 23: PRINT "TO CHANGE VALUES OF RHO,
     THETA, PHI,"
130  VTAB 24: PRINT "OR DISTANCE,
     PRESS 'RETURN'";
140  VTAB 24: HTAB 30: GET A$: VTAB 1: PRINT
150  IF ASC (A$) = 27 THEN TEXT : HOME : END
160  IF ASC (A$) < > 13 THEN 140
170  RETURN
180 RHO = 15:D = 350:THETA = .7:PHI = 1:
    CX = 140:CY = 96
190  DATA 1,1,1, 1,-1,1, -1,-1,1, -1,1,1
200  DATA 1,1,1, 1,1,-1, 1,-1,-1
210  DATA -1,-1,-1, -1,1,-1, 1,1,-1
220  DATA 1,-1,1, 1,-1,-1, -1,-1,1
230  DATA -1,-1,-1, -1,1,1,-1,1,-1
240  GOSUB 40: TEXT : HOME
250  VTAB 5: PRINT TAB( 10);"RHO = ";RHO
260  VTAB 8: PRINT TAB( 10);"THETA = ";THETA
270  VTAB 11: PRINT TAB( 10);"PHI = ";PHI
280  VTAB 14: PRINT TAB( 10);"DISTANCE = ";D
290  VTAB 20: PRINT "TO ACCEPT A VALUE,
     PRESS 'RETURN'"
300  VTAB 5: HTAB 15: INPUT R$
310  IF LEN (R$) < > 0 THEN RHO = VAL (R$)
320  VTAB 5: HTAB 15: PRINT " ";RHO;"        "
330  VTAB 8: HTAB 17: INPUT T$
340  IF LEN (T$) < > 0 THEN THETA = VAL (T$)
350  VTAB 8: HTAB 17: PRINT " ";THETA;"      "
360  VTAB 11: HTAB 15: INPUT P$
370  IF LEN (P$) < > 0 THEN PHI = VAL (P$)
380  VTAB 11:  HTAB 15: PRINT " ";PHI;" "
390  VTAB 14: HTAB 20: INPUT D$
400  IF LEN (D$) < > 0 THEN D = VAL (D$)
```

```
410   VTAB 14: HTAB 20: PRINT " "; D;" "
420   GOTO 240
430   TEXT : HOME : POKE 216,0
440   VTAB 10: PRINT "THE CUBE IS PARTLY
      OFFSCREEN."
450   VTAB 15: PRINT "TRY A LARGER VALUE OF
      RHO"
460   VTAB 20: PRINT "OR A SMALLER VALUE FOR
      DISTANCE"
470   FOR I = 1 TO 3000: NEXT I
480   HOME : GOTO 250
```







圖形的變換

前面提到的矩陣變換是用來建立點在標準和眼睛座標之間的關係。現在我們利用它來變換物體的點。

程式6.3產生三個不同的立體。行號120建立觀測點。行號190到300讀進DATA的資料並轉換成顯示幕座標，而把三個立體畫出來。第一個立體和程式6.1一樣。第二個和第三個則在顯示之前要先另外計算好。

立體2：

(a)把立體其Z軸的比例乘3。則變換矩陣爲

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b)把立體對Z軸旋軸30°，其變換矩陣爲

$$B = \begin{bmatrix} COS30° & SIN30° & 0 & 0 \\ SIN30° & COS30° & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} .866 & .5 & 0 & 0 \\ -.5 & .866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c)把立體的中心移到（－1，4，2）的地方，其變換矩陣爲

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 4 & 2 & 1 \end{bmatrix}$$

再把P點（X，Y，Z）連續變換成P′點（X1，Y1，Z1），利用（X1，Y1，Z1）＝（X，Y，Z，1）ABC

$$= \begin{bmatrix} .866 & .5 & 0 & 0 \\ -.5 & .866 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 1 & 4 & 2 & 1 \end{bmatrix}$$

因此

$$X1 = .866X - .5Y - 1$$

$$Y1 = .5X + .866Y + 4$$

$$Z1 = 3Z + 2$$

在程式6.3，行號60到90 計算變換的座標，並提供給行號100到110而得到顯示幕座標。

立體3：

(a)X和Y軸的比例因素爲2。

(b)把立體對X軸旋軸60°。

(c)把立體對Z軸旋軸15°。

(d)把立體的中心移到（4，2，－2）。

將a，b，c，d的變換矩陣相乘之後，我們得到這個矩陣

$$M=\begin{bmatrix} 1.93 & 0.518 & 0 & 0 \\ -.259 & .966 & 1.732 & 0 \\ .244 & -.836 & .5 & 0 \\ 4 & 2 & -2 & 1 \end{bmatrix}$$

結果，立體3其點的座標可寫爲

$$(X1,Y1,Z1,1)=(X,Y,Z,1)M$$

$$X1=1.93X-.259Y+.224Z+4$$

$$Y1=.518X+.966Y-.836Z+2$$

$$Z1=1.732Y+.5Z-2$$

行號20到50即在立體3得到顯示幕座標之前的變換。

程式6·3：變換立方體

```
1   REM PROGRAM 6.3 (TRANSFORMED CUBE)
2   REM DRAWS THREE IMAGES, GENERATED FROM A
    SINGLE DATA SET
10   GOTO 120
20 X1 = 1.93 * X - .259 * Y + .224 * Z + 4
30 Y1 = .518 * X + .966 * Y - .836 * Z + 2
40 Z1 = 1.732 * Y + .5 * Z - 2
50 X = X1:Y = Y1:Z = Z1: GOTO 100
60 X1 = .866 * X - .5 * Y - 1
70 Y1 = .5 * X + .866 * Y + 6
80 Z1 = 3 * Z + 2
90 X = X1:Y = Y1:Z = Z1
100 XE = - X * S1 + Y * C1:YE = - X * C1 *
    C2 - Y * S1 * C2 + Z * S2:ZE = - X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
110 SX = D * XE / ZE + CX:SY = CY - D * YE /
    ZE: RETURN
120 RHO = 20:D = 250:THETA = .3:PHI = 1.3:
    CX = 140:CY = 96:S1 = SIN (THETA):S2 = SIN
    (PHI): C1 = COS (THETA) :C2 = COS (PHI)
130   DATA 1,1,1, 1,-1,1, -1,-1,1, -1,1,1
140   DATA 1,1,1, 1,1,-1, 1,-1,-1
150   DATA -1,-1,-1, -1,1,-1, 1,1,-1
```

```
160   DATA 1,-1,1, 1,-1,-1, -1,-1,1
170   DATA -1,-1,-1, -1,1,1, -1,1,-1
180   HGR2 : HCOLOR= 3
190   FOR I = 1 TO 3: READ X,Y,Z
200   ON I GOSUB 100,60,20
210   HPLOT SX,SY
220   FOR J = 1 TO 11: READ X,Y,Z
230   ON I GOSUB 100,60,20
240   HPLOT TO SX,SY
250   NEXT J
260   FOR J = 1 TO 2: READ X,Y,Z
270   ON I GOSUB 100,60,20
280   HPLOT SX,SY: READ X,Y,Z
290   ON I GOSUB 100,60,20
300   HPLOT TO SX,SY
310   NEXT J: RESTORE : NEXT I
```

注意和建議

1. 修改程式 6.1以使其能產生其他的立體圖形，如角錐體，房屋，傢俱等等。每一個圖形，您心中必須先有個概略的模型，然後畫出草圖，再把要顯示的點製成表格。在目前，如果您想要消除應該隱藏的線和表面，只有把這些座標從表格中去掉。對每一種物體，我們都需要將各種不同的 RHO，THETA，PHI 和D值，來試驗之。

2. 把您寫的程式加上截割（clipping）的功能，這可以使用第四章所提到的方法，如此將可以允許圖形放大的能力，而不用擔心會超過顯示幕的範圍。

3. 我想您可能已經試過改變D值而使觀測者的位置介於物體和顯示幕之間（即用負的D值）。如果您還沒有在電腦上試過，是否可以預測其結果？如果觀測者的位置是在物體裏面，那又將如何呢？（在這種觀測位置時，被顯示的圖形一定要經過截割副程式的處理。）

Z = F（X，Y）形式的表面

我們已經發展出產生立體圖形的方法，進一步的我們要利用它來使表面（surfaces）產生多種變化。其程式的結構將和程式 6.3很

類似。我們必須

1. 提供表面上點的直角座標（X，Y，Z）；
2. 將之變換成顯示幕座標（SX, SY）；且
3. 畫出這些點

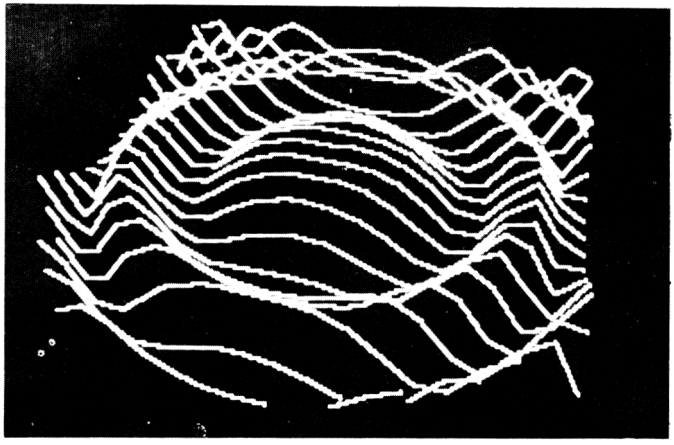
在此，程式並不像上一個程式用DATA來提供點的坐標，而是由程式中的某一部分計算而得。如下面這個程式：

這個程式所產生的圖形就是一個由一連串曲線所構成的表面，這些曲線則是由這個表面和平行於YZ平面的一些垂立平面的交點所構成。而我們就用這些平面曲線構成一幅立體圖形。這是一個典型的數學原理和電腦科學的例子：當您著手一個新問題時，將它化解成簡單的型式——一個您知道如何控制的型式。

由程式得到的圖形很明顯的是由行號50 的函數所決定。若要得到一個顯示表面重要特徵而令人喜愛之圖形，通常都是利用很費時的嘗試錯誤法（trial-and-error）來處理。稍微改變一下函數的定義，對圖形的影響就會很大。您可以把上面的函數稍加改變而試驗之

程式 6·4：表面

```
1   REM PROGRAM 6.4 (SURFACE)
2   REM DRAWS A SURFACE Z = F(X,Y)
10   GOTO 40
20 XE = - X * S1 + Y * C1:YE = - X * C1 * C2 - Y
   * S1 * C2 + Z * S2:ZE = - X * S2 * C1 - Y *
   S2 * S1 - Z * C2 + RHO
30 SX = D * XE / ZE + CX:SY = CY - D * YE / ZE:
   RETURN
40 RHO = 30:D = 350: THETA = .3:PHI = 1:CX =
   140:CY = 96:S1 = SIN (THETA): S2 = SIN
   (PHI):C1 = COS (THETA):C2 = COS (PHI)
50   DEF FN Z(X) = COS (.1 * (X * X + Y * Y))
60   HGR2 : HCOLOR= 3
70   FOR X = 10 TO - 10 STEP - 1
80 FL   = 0
90   FOR Y = - 10 TO 10
100 Z = FN Z(X): GOSUB 20
110   IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
      THEN FL = 0: GOTO 140
120   IF FL = 0 THEN FL = 1: HPLOT SX,SY
130   HPLOT   TO SX,SY
140   NEXT Y,X
```



。如

```
Z = 3*COS(.4*(X*X+Y*Y))
```

或

```
Z = COS(.1*(X*X+Y*Y)).
```

計算更多的點來連成曲線的話將會使表面更加平滑（即將行號90的STEP 值減少）。當然這樣的話將會增加執行的時間。

固然圖形是決定於函數，但也受觀測助變數RHO，THETA，PHI 和D值的影響很大。試試看RHO = 60 或PHI = 2，或您想要的值試驗之。

當選擇不同位置的觀測點，圖形很可能會因爲曲線的重疊而發生嚴重混淆，因爲程式是產生一個透明的表面，我們能看穿整個表面。有時候這是我們想要的，但通常它是一個問題，它帶來混淆的圖形和視覺的錯覺。若是我們能把圖形後面的部分隱藏起來，則這個圖形將更令人喜愛。對電腦繪圖而言，發展一套有效的隱藏線消去法這是一個最富挑戰性的問題。對這個問題，在下一章我們將加以探討。

把下面這幾個函數試驗看看，不過通常還要再調整一下觀測助變

數和範圍（ FOR-NEXT迴路的極限 ）。選擇一較大的STEP 值將使程式的執行更快，不過曲線也就較不圓滑。

1. `Z = 3*COS(.05*(X*X - Y*Y))`　　並試試看不同的觀測點。
2. `Z = 3*EXP(-.1*(X*X + Y*Y))`
3. `Z = A*SIN(X/B)*SIN(Y/C)`　　並試試看不同的A，B和C的值。
4. `Z = .1*(X*X - Y*Y)`

第七章　隱藏線和隱藏表面的處理

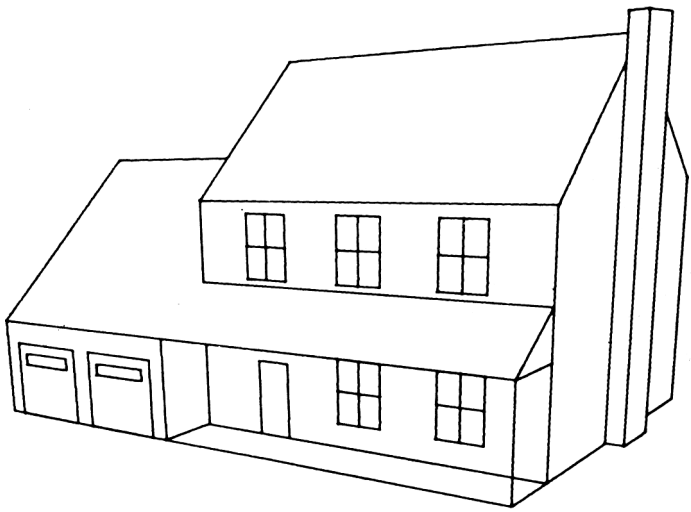


圖 7-1

緒論（ Introduction ）

速度比子彈還快！

力氣比火車還大！！

一躍即過摩天樓！！！

超人（SUPERMAN）另外還有X射線的透視眼；能夠看穿物體。而正常人在當我們費很大勁於機械設備中尋找損壞元件，或學習人體內部構造，或尋找汽車鑰匙時，或許會很羨慕透視眼的能力。但在另一方面，不受控制的X射線透視眼將造成不方便，一個X射線的世界，所有的東西都是透明或半透明，那將是一個非常混淆的世界，此時或許我們會想要能夠端詳一個物體，能夠把物體翻轉過來研究，而不希望看透整個物體。

以前我們所得到的立體物體都是屬於在一個X射線世界中，而我們的微電腦也就擁有了超人的透視眼。但我們卻寧願其動作更合於正常。如此則需要費一番努力來使它能受控制。我們必須發展一套方法來確認這些表面和線之中，有那些是可見的（或部分可見），而那些不是。然後才能把可見的部分顯示出來，而消去觀測不到的部分。

消除隱藏線和隱藏面是電腦繪圖最富挑戰性的問題之一。欲達此目的需要耗費電腦大量的時間和記憶體，而這些對電腦而言都是非常珍貴的。因此我們將限制我們的討論，僅提到一些技巧的動作原理，再提供一些畫出簡單立體物體的程式。因爲一個通用性質的隱藏線副程式，將會非常沒有效率，對設計者而言，使這些技巧僅侷限於特定用途將會有效率的多了。

測定表面的方位 （Surface Orientation）

從第六章的敍述，我們已經可以畫出立體物體，如果想要避免畫出物體的隱藏線和表面，則必須先要區別可見線和隱藏線。

若只要顯示物體的可見部分，一個很明顯的方法就是使電腦只定義物體的可見部分。程式6·1就是如此，它只把立方體的可見部分畫出來。程式只定義在指定觀測點下的可見表面頂點。這種技巧需要設計者先規劃想要的觀測點並據此而定義出物體的可見部分，此類程式都很短並且畫得很快。但無論如何它總是受限制的，如果觀測點改變的話，則程式的DATA陳述語和有關畫可見線的部分也都要再修改。

下面這個方法就比較適用些。程式判斷目前那個表面面對觀測點而認定它爲可見表面。

> 注意：下面的敍述是假設您已經精通向量（vectors）與其交積（cross products），點積（dot products）。這些概念在附錄3中也有一些簡要的介紹。

動作的原理

當我們看一個骰子時，可能有兩個或三個表面是可以讓我們看到的，而其他的則無法看見（如圖7.2）。底下我們將使用的第一個隱藏線技巧就是要利用到表面的方位或“面對方向”。

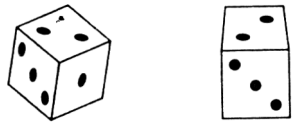


圖7-2　骰子的可見面

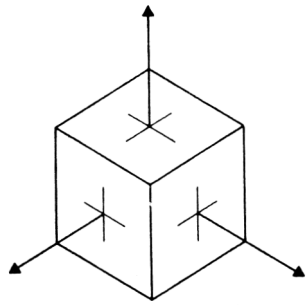


圖7-3　立方體的法線向量

一個立方體的每一面我們都可以得到兩個垂直方向，一個是向內，另一個則是向外。我們可用法線向量（ normal vectors ）來表示這些方向，並取向外的向量做爲表面的方位向量，如圖 7.3 所示。

另外在物體的每一個表面還可以得到第二個向量，即視線（line-of-sight ）向量。這是一個從表面的頂點（任何頂點）到觀測點（觀測者眼睛的位置）的向量。

對每一個表面我們都可以測量其視線向量和方位向量之間的角度。若角度在 0° 和 90°之間，則這個表面就是面對觀測者，故其爲可見並應被顯示。若角度在 90° 和 180° 之間，則此表面即背向觀測者，故爲不可見並且不應被顯示（如圖 7.4 ）。

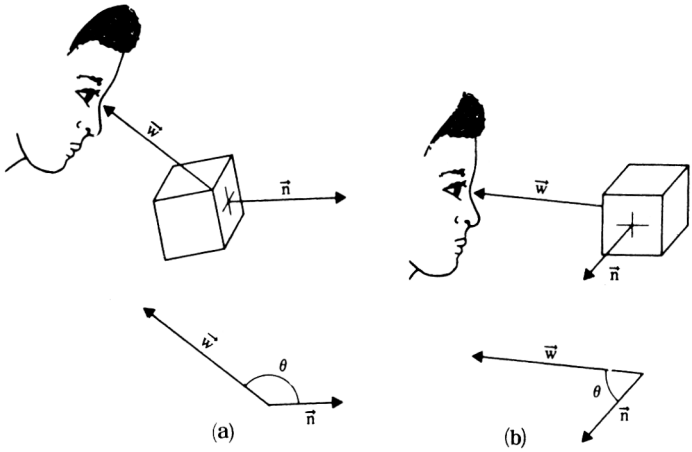


圖 7-4 視線向量和方位向量的關係

方位向量

在圖 7·5中，我們把立方體的一個表面上頂點編了號碼。第一個頂點是任意設定，但這表面其他頂點的號碼以反時針方向（從立方體外面並面對此表面觀測時）連續下去卻是重要的。接著，我們可以定義向量 u 。從頂點 1 到頂點 2 ，還有向量 v ，從頂點 1 到頂點 3 。則

交積（cross product），$\vec{n}=\vec{u}\times\vec{v}$，將會垂直立方體的表面，且方向向外。（至於方位向量 n 的大小，或長度，對我們並不重要。）

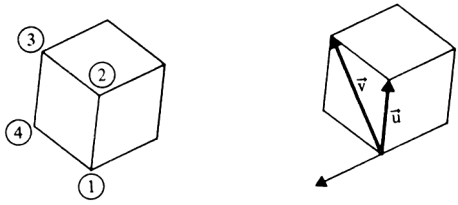


圖 7-5　方位向量

> 注意：表面上頂點的號碼以在立方體的外面，並面對表面時的反時針方向編成 1，2，3，4，這是很重要的，倘若編成順時針方向，則前面談到的向量 $\vec{n}=\vec{u}\times\vec{v}$，其方向將爲向內。

可見性（visibility）

向量 $\vec{w}$ 和 $\vec{n}$ 的點積（dot product）具有如下之性質

$$\vec{w}\cdot\vec{n}=|\vec{w}||\vec{n}|\cos\theta$$

在此 θ 爲 w 和 n 之間的角度，若 θ 在 0° 和 90° 之間，$\vec{w}\cdot\vec{n}$ 的值將爲正。若 θ 值在 90° 和 180° 之間，則 $\vec{w}\cdot\vec{n}$ 的值將爲負。

利用 $\vec{w}$ 來表示表面的視線向量，$\vec{n}$ 表示方位向量，我們可以做一個可見性的試驗。對一可見表面，如圖 7.4b，其 $\vec{w}$ 和 $\vec{n}$ 之間的角度將介於 0° 和 90° 之間，且 $\vec{w}\cdot\vec{n}$ 爲正。對一隱藏表面，如圖 7.4a，其 $\vec{w}$ 和 $\vec{n}$ 間的角度則介於 90° 和 180° 之間，且 $\vec{w}\cdot\vec{n}$ 爲負。

程式7.1

為了說明表面的方位在消去隱藏線時的用處，我們將討論一個畫出一間房屋的程式。（這個程式很長，所以我們分五個部分來說明，而完整的程式則列在後面。）

這個房屋是由7個表面所組成的，即兩個屋頂，一前壁，一後壁，兩個側面，和一個底面。而每個表面是由4或5個邊所組成。兩個"側面"都各有5個邊，其他的表面都是4個邊。而邊則依次由端點（或頂點）所組成。如圖7.6的房屋共有10個頂點。

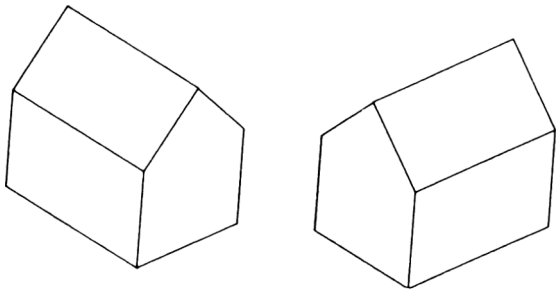


圖7-6 房屋的外觀

頂點陣列

我們從定義表面的10個頂點開始。為了參考之用我們把它編成號碼，如圖7.7所示。

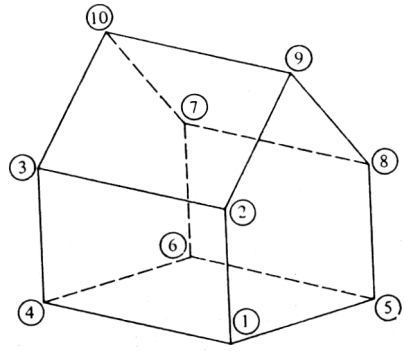


圖7-7 房屋頂點的編號

頂　點	座		標
1	5	7	−5
2	5	7	5
3	5	−7	5
4	5	−7	−5
5	−5	7	−5
6	−5	−7	−5
7	−5	−7	5
8	−5	7	5
9	0	7	8
10	0	−7	8

程式 7·1：第一部分

```
10  REM VIEWPOINT AND SCREEN PARAMETERS
20 RHO = 40:THETA = .7:PHI = 1:D = 250
30 CX = 140:CY = 96:S1 = SIN (TH):C1 = COS
   (TH):S2 = SIN (PH):C2 = COS (PH)
40  REM VERTICES
50  DATA 5,7,-5
60  DATA 5,7,5
70  DATA 5,-7,5
80  DATA 5,-7,-5
90  DATA -5,7,-5
100  DATA -5,-7,-5
110  DATA -5,-7,5
120  DATA -5,7,5
130  DATA 0,7,8
140  DATA 0,-7,8
150  REM FILL VERTEX ARRAYS
160  DIM V(10,3),SV(10,2)
170  FOR I = 1 TO 10
180  READ X,Y,Z
190 V(I,1) = X:V(I,2) = Y:V(I,3) = Z
200 X1 = - X * S1 + Y * C1:Y1 = - X * C1 * C2 -
    Y * S1 * C2 + Z * S2:Z1 = - X * S2 * C1 - Y
    * S2 * S1 - Z * C2 + RHO
210 SV(I,1) = D * (X1 / Z1) + CX:SV(I,2) = - D *
     (Y1 / Z1) + CY
220  NEXT I
```

程式的行號 40 到 220 是把頂點座標放在一個二階的頂點陣列。對每個陣列V（ I ，J ）而言，I 爲頂點的編號（ 1 到 10 之間），J 爲辨認座標的號碼。例如，V（ 8 ，1 ）爲第 8 個頂點的第一個座標（ 卽X座標），V（ 8 ，2 ）爲第 8 個頂點的第二個座標（ 卽Y座標），而V（ 8 ，3 ）則爲第 8 個頂點的第三個座標（ 卽Z座標）。因爲第 8 個頂點是（ －5 ，7 ，5 ），故

V（ 8 ，1 ）＝－ 5
V（ 8 ，2 ）＝ 7
V（ 8 ，3 ）＝ 5

當頂點座標放進頂點陣列後，每一個點的顯示幕座標卽被計算爲顯示幕頂點陣列SV（I，J），對於每一個 I ，SV（I，1）和SV（I，2）分別爲第 I 點的顯示幕座標SX，SY 。

表面陣列

在這個物體（ 房屋）上總共有 7 個表面要被畫。表面的編號如圖 7.8 。

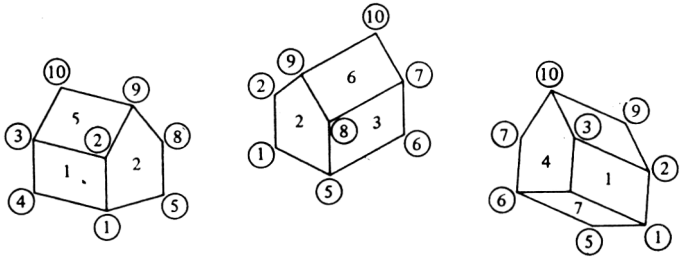


圖 7-8 房屋表面和頂點的編號

若連接適當的頂點，我們就能畫出一個表面，如：對表面 1 ，連接頂點 1 到 2 到 3 到 4 到 1 ；對表面 2 ，連接頂點 1 到 5 到 8 到 9 到 2 到 1 。其他表面則依此類推（ 請參閱圖 7.9 ）。

陣列S（ I ，J ）乃對每個表面 I ，指出其頂點而定義此表面。

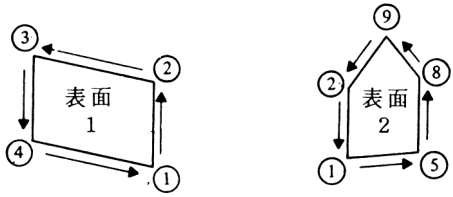


圖 7-9　表面的形成

因爲表面 1 是連接頂點 1 ，2 ，3 ，4 ，和 1 ，故這些號碼將分別存到 S(1,1)，S(1,2)，S(1,3)，S(1,4) 和 S(1,5) 。同樣的，S(2,1) = 1 ，S(2,2) = 5 ，S(2,3) = 8 ，S(2,4) = 9 ，S(2,5) = 2 和 S(2,6) = 1 。

完整的陣列如下：

		J（被連接的點）					
	S(I,J)	1	2	3	4	5	6
	1	1	2	3	4	1	0
	2	1	5	8	9	2	1
I	3	5	6	7	8	5	0
（表面）	4	4	3	10	7	6	4
	5	3	2	9	10	3	0
	6	7	10	9	8	7	0
	7	1	4	6	5	1	0

每一個表面的頂點以從物體外面看進去的反時針方向之順序排列，這是非常重要的。您應該參考圖 7.8仔細查對每一個點是否以適當的順序放入表面陣列裏。

程式 7.1 ：第二部分

```
230 REM NUMBER OF POINTS IN EACH SURFACE
240 DATA 5,6,5,6,5,5,5
250 DIM NPS(7)
260 FOR I = 1 TO 7
270 READ NPS(I)
280 NEXT I
290 REM SURFACE ARRAY - POINTERS TO VERTICES
300 DATA 1,2,3,4,1
310 DATA 1,5,8,9,2,1
320 DATA 5,6,7,8,5
330 DATA 4,3,10,7,6,4
340 DATA 3,2,9,10,3
350 DATA 7,10,9,8,7
360 DATA 1,4,6,5,1
370 REM FILL SURFACE ARRAY
380 DIM S(7,6)
390 FOR I = 1 TO 7
400 FOR J = 1 TO NPS(I)
410 READ S(I,J)
420 NEXT J,I
```

法線陣列

欲達成消除隱藏線所需之表面方位測定，需要一個垂直表面的向量。這些由程式中的行號430到550所計算。

程式7·1：第三部分

```
430  REM CALCULATE VECTOR NORMAL TO EACH SURFACE
440  DIM N(7,3)
450  FOR I = 1 TO 7
460 U1 = V(S(I,2),1) - V(S(I,1),1)
470 U2 = V(S(I,2),2) - V(S(I,1),2)
480 U3 = V(S(I,2),3) - V(S(I,1),3)
490 V1 = V(S(I,3),1) - V(S(I,1),1)
500 V2 = V(S(I,3),2) - V(S(I,1),2)
510 V3 = V(S(I,3),3) - V(S(I,1),3)
520 N(I,1) = U2 * V3 - V2 * U3
530 N(I,2) = U3 * V1 - V3 * U1
540 N(I,3) = U1 * V2 - V1 * U2
550  NEXT I
```

在發展法線陣列時，表面陣列內，點的順序是很重要的。這個陣列存著朝向表面外的法線向量之座標。N(1,1)，N(1,2)和N(1,3)即表面1的法線向量之X，Y，Z分量。N(2,1)，N(2,2)

和N(2,3)則爲表面2的法線向量之分量。對每一個表面I（$1 \leq I \leq 7$），陣列N(I,1)，N(I,2)和N(I,3)的值卽爲垂直此表面且向外的向量之分量。

法線向量的計算利用到一個定理，卽兩向量之交積會垂直此兩向量的平面。爲確認一向量垂直表面，我們首先選擇兩個平行於表面的向量$\vec{u}$和$\vec{v}$，然後$\vec{u} \times \vec{v}$就會產生一個垂直向量。

每一個表面有5或6個頂點，程式的行號460到510　定義$\vec{u}$爲從這些頂點的第一個延伸到第二個，$\vec{v}$則爲從第一個延伸到第三個。行號520到540是計算$\vec{u}$和$\vec{v}$點積之分量。

注意頂點的順序將如何影響法線向量的計算。如果表面的頂點是以順時針方向排列，則$\vec{u}$和$\vec{v}$對應的方位將會相反，且$\vec{u} \times \vec{v}$的結果將是一個朝向物體裏的法線向量，而不是向外的法線向量（如圖7.10）。

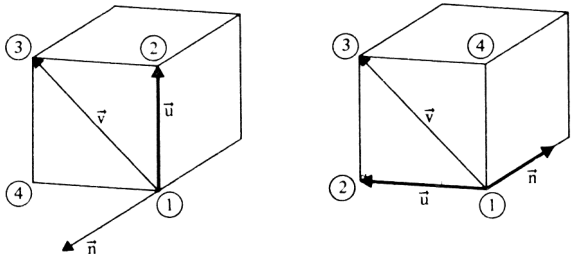


圖7-10　頂點順序與法線向量的關係

可見性的測試

現在我們就準備爲物體的每一個表面做可見性的測試。每一個表面，我們選擇一個從表面上一點（任何一點）延伸到觀測點（XE，YE,ZE）的視線向量$\vec{w}$。而$\vec{w}$和向外的法線$\vec{n}$之間的角度，提供給我們做爲判斷可見性的標準。若角度在0°和90°之間，表面就面對觀測者，且點積$\vec{n} \cdot \vec{w}$爲正。若角度在90°和180°之間，則表面就

背對觀測者，且n・w爲負。

程式的行號650就是可見性的過濾器。凡面對觀測者的表面（n・w＞0）將通過這個過濾器，且在以後會被畫出來。而沒有面對觀測者的表面（n・w≤0）將被拒絕而不理會。

程式7·1：第四部分

```
560  REM DETERMINE VISIBILITY OF EACH SURFACE
570  REM AND COLLECT VISIBLE EDGES IN EDGE ARRAY
580 XE = RH * S2 * C1:YE = RH * S2 * S1:
    ZE = RH * C2
590  DIM E(12,3):N = 1
600  FOR I = 1 TO 7
610 E2 = S(I,1)
620 WX = XE - V(E2,1)
630 WY = YE - V(E2,2)
640 WZ = ZE - V(E2,3)
650  IF N(I,1) * WX + N(I,2) * WY + N(I,3) *
     WZ < = 0 THEN 760
660 E1 = S(I,1)
670  FOR J = 2 TO NPS(I)
680 E2 = S(I,J)
690  FOR K = 1 TO N
700  IF E(K,1) = E2 AND E(K,2) = E1
     THEN E(K,3) = 2: GOTO 740
710  NEXT K
720 E(N,1) = E1:E(N,2) = E2:E(N,3) = 1
730 N = N + 1
740 E1 = E2
750  NEXT J
760  NEXT I
```

邊陣列

可見表面可以把它分解成邊而集中在已經被宣告而可接受12個邊的邊陣列裏〔DIM E(12,3)〕。

每一個被放在陣列裏的邊由一編號I（介於1和2之間）來辨認。對每個邊I，我們要存進三個數字。E(I,1)和E(I,2)爲構成此邊的兩個端點，E(I,3)則是一個標記（tag），用來確認以

此邊爲界限的可見表面之數目。若此邊是一個可見表面的界限，則E（Ｉ，３）的值爲１，若是兩個可見表面的界線，則E（Ｉ，３）爲２。

圖7.11顯示了在邊陣列裏，邊的分類名稱。

邊	端	點	標記
1	1	2	2
2	2	3	2
3	3	4	1
4	4	1	1
5	1	5	1
6	5	8	1
7	8	9	1
8	9	2	2
9	9	10	1
10	10	3	1
11	0	0	0
12	0	0	0

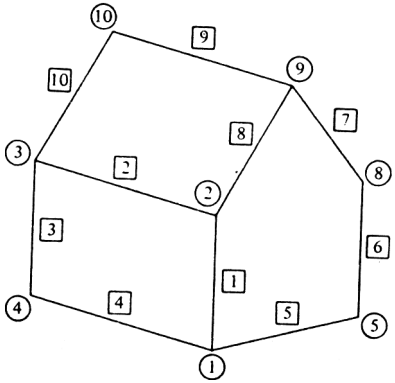


圖 7-11　邊的編號

在此時的觀測，第11，12個邊是不需要的。這個陣列提供的空間比需要的還多；沒有一種觀察角度會超過12個邊被顯示。

注意被畫在外面的邊（第3，4，5，6，7，9，10個邊）其標記的值皆爲1，這些邊都僅是一個可見表面的界限。而畫在裏面的邊（1，2，8）則爲兩個可見表面的界限，他們的標記值都等於2。“內邊”和“外邊”的區別在我們畫多個物體時就變得很重要了。

程式的行號660到760把邊陣列填滿，每個邊的端點是從表面陣列裏讀進來的（行號660，680，和740）並存進邊陣列裏（行號720），且標記值設爲1。若一個邊被碰見第二次，則其標記值卽設定爲2（行號700）。

畫圖

現在該是把可見邊畫出來的時候了，行號790到830先找出可見邊（標記值不爲零者），再讀進其對應之端點的編號，然後畫出來。

程式7·1：第五部分

```
770   REM DRAW VISIBLE EDGES
780   HGR2 : HCOLOR= 3
790   FOR I = 1 TO 12
800   IF E(I,3) = 0 THEN 830
810 J = E(I,1):K = E(I,2)
820   HPLOT SV(J,1),SV(J,2) TO SV(K,1),SV(K,2)
830   NEXT I
840   END
```

程式7·1：消除隱藏線（測定表面方位法）

```
1   REM PROGRAM 7.1 (HIDDEN LINE ELIMINATION)
2   REM SURFACE ORIENTATION METHOD
10   REM VIEWPOINT AND SCREEN PARAMETERS
20 RHO = 40:THETA = .7:PHI = 1:D = 250
30 CX = 140:CY = 96:S1 = SIN (TH):C1 = COS (TH):
```

```
   S2 = SIN (PH):C2=COS (PH)
40   REM VERTICES
50   DATA 5,7,-5
60   DATA 5,7,5
70   DATA 5,-7,5
80   DATA 5,-7,-5
90   DATA -5,7,-5
100   DATA -5,-7,-5
110   DATA -5,-7,5
120   DATA -5,7,5
130   DATA 0,7,8
140   DATA 0,-7,8
150   REM FILL VERTEX ARRAYS
160   DIM V(10,3),SV(10,2)
170   FOR I = 1 TO 10
180   READ X,Y,Z
190 V(I,1) = X:V(I,2) = Y:V(I,3) = Z
200 X1 = -X * S1 + Y * C1:Y1 = -X * C1 *
    C2 -Y * S1 * C2 + Z * S2:Z1 = -X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
210 SV(I,1) = D * (X1 / Z1) + CX:SV(I,2) = -D *
    (Y1 / Z1) + CY
220   NEXT I
230   REM NUMBER OF POINTS IN EACH SURFACE
240   DATA 5,6,5,6,5,5,5
250   DIM NPS(7)
260   FOR I = 1 TO 7
270   READ NPS(I)
280   NEXT I
290   REM SURFACE ARRAY - POINTERS TO VERTICES
300   DATA 1,2,3,4,1
310   DATA 1,5,8,9,2,1
320   DATA 5,6,7,8,5
330   DATA 4,3,10,7,6,4
340   DATA 3,2,9,10,3
350   DATA 7,10,9,8,7
360   DATA 1,4,6,5,1
370   REM FILL SURFACE ARRAY
380   DIM S(7,6)
390   FOR I = 1 TO 7
400   FOR J = 1 TO NPS(I)
410   READ S(I,J)
420   NEXT J,I
430   REM CALCULATE VECTOR NORMAL TO EACH SURFACE
440   DIM N(7,3)
450   FOR I = 1 TO 7
460 U1 = V(S(I,2),1) - V(S(I,1),1)
470 U2 = V(S(I,2),2) - V(S(I,1),2)
480 U3 = V(S(I,2),3) - V(S(I,1),3)
490 V1 = V(S(I,3),1) - V(S(I,1),1)
500 V2 = V(S(I,3),2) - V(S(I,1),2)
```

(continued)

```
510 V3 = V(S(I,3),3) - V(S(I,1),3)
520 N(I,1) = U2 * V3 - V2 * U3
530 N(I,2) = U3 * V1 - V3 * U1
540 N(I,3) = U1 * V2 - V1 * U2
550  NEXT I
560  REM DETERMINE VISIBILITY OF EACH SURFACE
570  REM AND COLLECT VISIBLE EDGES IN EDGE ARRAY
580 XE = RH * S2 * C1:YE = RH * S2 * S1:
    ZE = RH * C2
590  DIM E(12,3):N = 1
600  FOR I = 1 TO 7
610 E2 = S(I,1)
620 WX = XE - V(E2,1)
630 WY = YE - V(E2,2)
640 WZ = ZE - V(E2,3)
650  IF N(I,1) * WX + N(I,2) * WY + N(I,3) * WZ
     < = 0 THEN 760
660 E1 = S(I,1)
670  FOR J = 2 TO NPS(I)
680 E2 = S(I,J)
690  FOR K = 1 TO N
700  IF E(K,1) = E2 AND E(K,2) = E1
     THEN E(K,3) = 2: GOTO 740
710  NEXT K
720 E(N,1) = E1:E(N,2) = E2:E(N,3) = 1
730 N = N + 1
740 E1 = E2
750  NEXT J
760  NEXT I
770  REM DRAW VISIBLE EDGES
780  HGR2 : HCOLOR= 3
790  FOR I = 1 TO 12
800  IF E(I,3) = 0 THEN 830
810 J = E(I,1):K = E(I,2)
820  HPLOT SV(J,1),SV(J,2) TO SV(K,1),SV(K,2)
830  NEXT I
840  END
```

注意和建議

1 或許您會對上面的程式感到不耐煩，並知道一些更有效率，更好的方法。這個程式是被設計成以一種容易被敍述的方式來說明如何利用表面方位達成消除隱藏線。並介紹一些技巧（如邊陣列的標記）.，這在以後是很重要的。因此，其目標並非完成一最有效率的程式。

例如，一階的陣列就此二階陣列更快被儲取。

2. 觀測點助變數在行號20 被指定，試改變一下觀測的位置。通常，您會只想改變THETA或PHI 。

3. 選擇一較小的RHO 值或較大的D值，可能造成一不適合顯示幕的圖形，或許您會想在行號750插入一個截割程式（請看第四章）。

4. 若欲畫其他的凸面多邊體時，很明顯地，將需要更改頂點陣列和表面陣列的資料，並更改有關的FOR-NEXT 廻路的極限值。試試如立方體，角錐體，和角柱體等。

5. 只要一個表面被確認爲可見的（面對觀測者），則此表面上的其他元素亦皆爲可見。在房屋四週的每一面牆上，加上窗戶和門，則每當其所在之表面爲可見時，把這些元素也畫出來。

兩個物體；塗黑法 （Blackout）

上面所講的方法只能畫單一物體的可見部分。底下我們要講的方法，則允許畫多個物體的可見部分。除了要考慮物體的那些部分面對或背對觀測者之外，我們還必須確認那些部分被較近的物體所隱藏。

當我們環顧四週，物體的某些部分會由於光不能穿透非透明體，以致被靠我們較近的物體所遮住。我們可以把它看成已經被“畫”，但又被較近物體的“畫”所“消除”（看圖7.12）。如程式7.2即是。

程式7·2：消除隱藏線（兩個物體；塗黑法）

```
1  REM PROGRAM 7.2 (HIDDEN LINE ELIMINATION)
2  REM TWO OBJECTS; BLACKOUT
10  TEXT : HOME : GOTO 130
20 XE = -X * S1 + Y * C1:YE = -X * C1 * C2 - Y *
   S1 * C2 + Z * S2:ZE = -X * S2 * C1 - Y * S2 *
   S1 - Z * C2 + RHO
```

```
30 SX = D * XE / ZE + CX:SY = CY - D * YE / ZE:
   RETURN
40  DATA 1,1,0, 0,0,1, 0,1,0, 1,1,0, 1,0,0,
    0,0,1
50  DATA 1,1.5,1, 0,1.5,1
60  DATA 0,2.5,1, 1,2.5,1, 1,2.5,0,
    0,2.5,0,0,2.5,1, 0,1.5,1, 1,1.5,1, 1,1.5,0,
    1,2.5,0,1,2.5,1, 1,1.5,1
70  FOR I = 1 TO N: READ X,Y,Z
80  GOSUB 20
90  IF I = 1 THEN HPLOT SX,SY
100  HPLOT TO SX,SY
110  NEXT I
120  RETURN
130 RHO = 10:D = 600:THETA = 1:PHI = 1.2:
    CX = 140: CY = 96
140 S1 = SIN (THETA):C1 = COS (THETA)
150 S2 = SIN (PHI):C2 = COS (PHI)
160  HGR2 : HCOLOR= 7
170 N = 6: GOSUB 70
180  READ X,Y,Z: GOSUB 20:X1 = SX:Y1 = SY
190  READ X,Y,Z: GOSUB 20:X2 = SX:Y2 = SY
200 SL = (Y1 - Y2) / (X1 - X2)
210  HCOLOR= 4
220  FOR XP = X1 TO X2:YP = SL * (XP - X1) + Y1
230  HPLOT XP,YP TO XP,191
240  NEXT XP
250  HCOLOR= 7
260 N = 9: GOSUB 70:N = 2: GOSUB 70
```

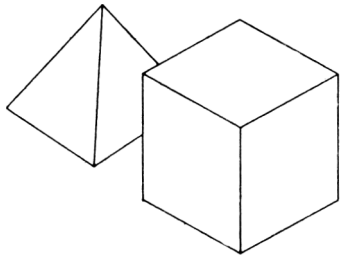


圖 7-12 立方體和角錐體

行號 40 到 60 的資料爲一個立方體和角錐體（如圖 7.12 ）的頂點，但只有面對觀測者的部分。兩個物體中，較遠的一個（角錐體）先被畫（行號 170 ），在畫較近物體（立方體）之前把角錐體應該被隱藏部分先從顯示幕移走，這可以使用 HCOLOR = 4（背景的顏色，黑色）來畫被立方體佔據的顯示幕部分。行號 180 到 240 即控制角錐

體隱藏部分的消除。從立方體的一個邊畫一條黑色垂直線到顯示幕底部。完成了消除，則立方體即被畫（行號 260 ）。

這種消除隱藏線的技巧，非常容易使用，並在很短的執行時間內得到一個很好的結果。不過它卻有些限制；觀測點必須事先被規劃好，以得到可見表面的頂點，並找出被“塗黑”的地區，因此若觀測點改變則亦須改變DATA陳述語和塗黑的程式。

兩個物體；二分找尋法 （Binary Seach）

圖 7.13 中，在觀測時，立方體應該擋住角錐體的頂點H，角錐體的VH邊應該部分被隱藏，而只有VT 爲可見。若已知頂點V的顯示幕座標，則必須算出T點的座標，因爲T是角錐體邊VH 和立方體邊PQ的交點，而T的座標可以由線VH和PQ的方程式求得，但此法比較沒有效率，因此我們推薦一種“二分找尋法”來得到T的座標。

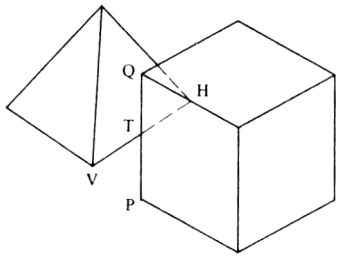


圖 7-13

程式 7·3：消除隱藏線（兩個物體；二分找尋法）

```
1  REM PROGRAM 7.3 (HIDDEN LINE ELIMINATION)
2  REM TWO OBJECTS; BINARY SEARCH
10  TEXT : HOME : GOTO 140
20 XE = -X * S1 + Y * C1:YE = -X * C1 * C2 - Y *
   S1 * C2 + Z * S2:ZE = -X * S2 * C1 - Y * S2 *
   S1 - Z * C2 + RHO
30 SX = D * XE / ZE + CX:SY = CY - D * YE / ZE:
   RETURN
40  DATA 0,2.5,1, 1,2.5,1, 1,2.5,0,
    0,2.5,0,0,2.5,1, 0,1.5,1, 1,1.5,1, 1,1.5,0,
    1,2.5,0,1,2.5,1, 1,1.5,1
```

```
50  DATA 1,1,0, 0,0,1, 1,0,0, 1,1,0
60  DATA 1,1,0, 0,1,0, 1,1.5,0, 1,1.5,1
70  DATA 0,0,1, 0,1,0, 1,1.5,1, 0,1.5,1
80  FOR I = 1 TO N: READ X,Y,Z
90  GOSUB 20
100  IF I = 1 THEN HPLOT SX,SY
110  HPLOT TO SX,SY
120  NEXT I
130  RETURN
140 RHO = 10:D = 600:THETA = 1:PHI = 1.2:
    CX = 140:CY = 96
150 S1 = SIN (THETA):C1 = COS (THETA)
160 S2 = SIN (PHI):C2 = COS (PHI)
170  HGR2 : HCOLOR= 7
180 N = 9: GOSUB 80:N = 2: GOSUB 80
190 N = 4: GOSUB 80
200  FOR J = 1 TO 2
210  READ X,Y,Z: GOSUB 20:XV = SX:YV = SY
220  READ X,Y,Z: GOSUB 20:XH = SX:YH = SY
230  READ X,Y,Z: GOSUB 20:XP = SX:YP = SY
240  READ X,Y,Z: GOSUB 20:XQ = SX:YQ = SY
250 V1 = (XH - XV) / 2:V2 = (YH - YV) / 2
260 U1 = XP - XQ:U2 = YP - YQ
270 XT = XV + V1:YT = YV + V2
280  FOR I = 2 TO 7
290 VI = SGN ((U2 * (XV - XQ) - U1 *
    (YV - YQ)) * (U2 * (XT - XQ) - U1 *
    (YT - YQ)))
300 V1 = V1 / 2:V2 = V2 / 2
310 XT = XT + VI * V1:YT = YT + VI * V2
320  NEXT I
330  HPLOT XV,YV TO XT,YT
340  NEXT J
```

行號180和190是畫立方體的可見部分和角錐完全可見的邊。行號210到240在得到點V，H，P和Q的顯示幕座標。T點座標的計算則在行號250到320，並使用二分找尋法，如下：

1. 得到向量V(V1,V2)，從點V(XV,YV)延伸到點H(XH,YH)的一半。(行號250)
2. 把(XV,YV)加上(V1,V2)而得到從(XV,YV)到(XH,YH)的線段之中點(XT,YT)。(行號270)
3. 測試(XT,YT)的可見性。(行號290)
4. 重新定義(V1,V2)=(V1,V2)/2。(行號300)

5. 如果（XT，YT）爲可見，則重新定義（XT，VT）＝（XT，VT）＋（V1，V2），而向（XH，YH）移動。若（XT，YT）爲隱藏，則重新定義（XT，YT）＝（XT，YT）－（V1，V2），而向（XV，YV）移動。（行號310）
6. 回到第三步。（行號320）

爲了要決定T的可見性，我們利用從Q點到P點的向量$\vec{u}$，和從Q點到V點的向量（XV－XQ，YV－YQ），以及從Q點到T點的向量（XT－XQ，YT－YQ）。若T是可見的話，則T一定位於線PQ和V點這一邊，而（XT－XQ，YT－YQ）和$\vec{u}$的點積與（XV－XQ，YV－YQ）和$\vec{u}$的點積必有相同的符號。若T爲不可見，則兩個點積符號相反。（我們可以放心的假設沒有一個點積爲零嗎？）行號290在處理這些點積的比較。

（XT，YT）在其第七個值被計算過後即可確定。此時（V1，V2）的長度爲（XV，YV）和（XH，YH）之間距離的1／280，對大部分的APPLE顯示幕圖形，這是可接受的精確度。而更大的精確度可由增加廻路的週期而得。

在離開這個廻路時，V點到T點的線段就被畫出來。然後又對第二個的部分隱藏線重複上述的程序。

兩個物體；任意的觀測點（Arbitrary Viewpoint）

前面兩個方法會受制於觀測點，若想要更通用的方法，我們可以擴充表面方位法（surface orientation method）。

動作原理

我們將利用圖7.14的物體來說明一些必要的計算。在7.14a的觀測角度，立方體較接近觀測者，並隱藏一部分的角錐體。在7.14b中則是角錐體隱藏了部分的立方體。每一個物體的優先順序將是我們的考慮之一。當兩物體在一起時，沒有被另外一個物體所隱藏的將是

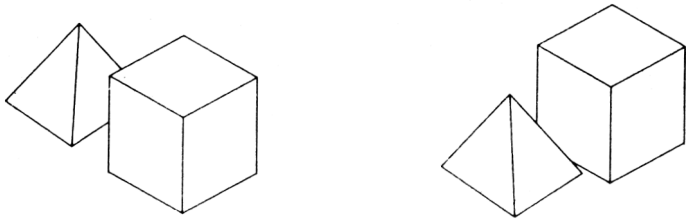


圖 7-14

第一優先，而部分被隱藏的物體則被指定爲第二優先。優先順序的指定可以由從觀測點到物體中央的距離來判斷，較近的物體就被指定爲第一優先。

指定優先順序的第二個方法是利用“隔離面（ separating plane ）”（ 隔離兩個物體的面 ）。與觀測點在隔離面同一邊的物體卽爲第一優先。任何能分離兩個物體的平面皆可做爲“隔離面”（ 此隔離可以包含一個或兩個物體的點，或邊，或表面。）。如果隔離面的改變會改變物體的優先順序時，我們不用擔心，在這種情況下，沒有一個物體會隱藏其他的物體。

一旦物體的優先順序被指定好，我們就畫出第一優先的物體，並利用表面方位的測定只畫出面對觀測者的表面。

第二優先的物體也將用到表面方位的處理，來找出面對觀測者的表面。然後一個可見表面的每一邊就必須被個別考慮以決定它的兩個端點是否被第一優先的物體所隱藏。如果沒有端點被隱藏，則此邊就被棄置不管。若只有一個端點被隱藏，我們必須計算此邊的顯示圖形與較優先物體其邊界的顯示圖形之交點，然後畫出此邊的可見部分。

這個問題呈現出以前從未面臨過的挑戰。我們將以一個立方體和角錐體的一個邊來說明。

圖 7.15a中，角錐體其兩個邊之隱藏部分以虛線表示。因爲這兩個邊的可見部分都是從其可見端點延伸到其與立方體圖形之外邊界（

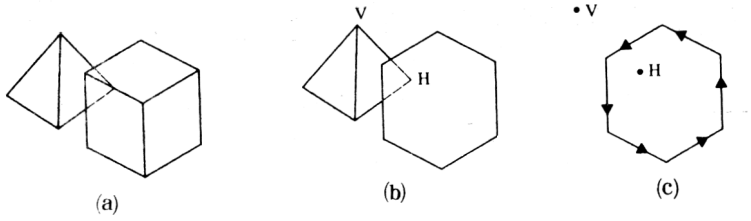


圖 7-15

outside boundary ）的交點。故我們可以去掉立方體的內部邊界而簡化圖形（如圖 7.15b ）。

在圖 7.15c中，立方體外邊的方位乃承自表面陣列記入時的定義。在這種畫法，注意角錐體邊的H點是在立方體邊界內部，因此被立方體隱藏，並且也在立方體每一個邊界的左半面（ left half-plane ）。V點，並沒有被立方體所隱藏，且位於立方體某些邊界的左半面，及其他邊界的右半面。（請看附錄 3 對半面（ half-plane ）的說明。）

爲要確認角錐體其邊的可見部分，我們沿著從H到V的線段移動，並測試每一點的可見性，當發現第一個位於至少一個立方體邊界之右半平面的點，即爲T點。則V到T的可見線段，就可以被畫出來。

程式7.4

以上所講的方法我們可以用下面這個程式來說明。對於程式 7.1 的“房屋”，我們再加上第二個物體：“煙囪”。（同樣的，這個程式也很長，我們將把它分成幾個部分來說明。）

這個煙囪由 6 個表面組成：頂面，底面，和四個側面。而每個表面是由四個邊所組成，每個邊由兩個端點組成。

在定義這兩個物體時，房屋的頂點是用程式 7.1已經定義好的。（圖 7.16 ）

至於煙囪的頂點，則將其編成號碼，如圖 7.17 。

頂點	座	標	
1	5	7	−5
2	5	7	5
3	5	−7	5
4	5	−7	−5
5	−5	7	−5
6	−5	−7	−5
7	−5	−7	5
8	−5	7	5
9	0	7	8
10	0	−7	8

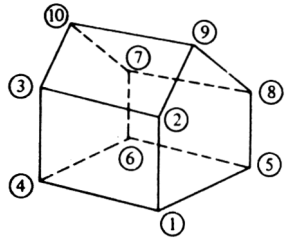


圖 7-16 房屋頂點的編號

頂點	座	標	
11	−2	−7	−5
12	−2	−9	−5
13	−5	−9	−5
14	−5	−7	−5
15	−2	−7	9
16	−2	−9	9
17	−5	−9	9
18	−5	−7	9

頂點陣列

程式的行號10到310，在確認觀測點和顯示幕助變數，並把頂點座標讀進頂點陣列裏。陣列V（I，J）和SV(I，J)被宣告成能容納18個點的座標。頂點陣列V（I，J）的作用和以前一樣，用V（I，1），V（I，2），和V（I，3）來存頂點I的X，Y，和Z座標。

程式7·4：第一部分

```
10   REM VIEWPOINT AND SCREEN PARAMETERS
20 RHO = 40:THETA = .7:PHI = 1:D = 250
30 CX = 140:CY = 96:S1 = SIN (TH):
   C1 = COS (TH):S2 = SIN (PH):C2 = COS (PH)
40   REM VERTICES OF OBJECT 1 (HOUSE)
50   DATA 5,7,-5
60   DATA 5,7,5
70   DATA 5,-7,5
80   DATA 5,-7,-5
90   DATA -5,7,-5
100   DATA -5,-7,-5
110   DATA -5,-7,5
120   DATA -5,7,5
130   DATA 0,7,8
140   DATA 0,-7,8
150   REM VERTICES OF OBJECT 2 (CHIMNEY)
160   DATA -2,-7,-5
170   DATA -2,-9,-5
180   DATA -5,-9,-5
190   DATA -5,-7,-5
200   DATA -2,-7,9
210   DATA -2,-9,9
220   DATA -5,-9,9
230   DATA -5,-7,9
240   REM FILL VERTEX ARRAYS
250   DIM V(18,3),SV(18,3)
260   FOR I = 1 TO 18
270   READ X,Y,Z
280 V(I,1) = X:V(I,2) = Y:V(I,3) = Z
290 X1 = -X * S1 + Y * C1:Y1 = -X * C1 * C2 - Y *
    S1 * C2 + Z * S2:Z1 = -X * S2 * C1 - Y * S2 *
    S1 - Z * C2 + RHO
300 SV(I,1) = D * (X1 / Z1) + CX:
    SV(I,2) = -D * (Y1 / Z1) + CY
310   NEXT I
```

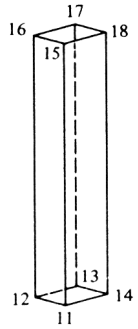


圖 7-17 煙囪頂點的編號

SV(I,1)和SV(I,2)是用V(I,1),V(I,2)和V(I,3)的值計算出來的。對每個I而言，SV(I,1)和SV(I,2)爲點(X,Y,Z)的顯示幕座標。另外，每一個頂點I還可以得到SV(I,3)，在此每一個SV(I,3)等於起始值0，但以後將有所更改，以表示某些頂點的狀態(可見或隱藏)。

表面陣列

接著是表面陣列，此陣列其實就是擴大在程式7.1所使用的陣列。表面1到7做爲房屋的表面，而8到13則是煙囪的表面。表面8和9是煙囪的頂面和底面；而10，11，12，和13則爲其側面。和以前一樣，陣列V(I,1)到V(I,NPS(I))以反時針方向(從物體外面看進去)的順序存進表面I的頂點。

程式7·4：第二部分

```
320   REM NUMBER OF POINTS IN EACH SURFACE
330   DATA 5,6,5,6,5,5,5,5,5,5,5,5,5
340   DIM NPS(13)
```

```
350  FOR I = 1 TO 13
360  READ NPS(I)
370  NEXT I
380  REM SURFACE ARRAY - POINTERS TO VERTICES
390  DATA 1,2,3,4,1
400  DATA 1,5,8,9,2,1
410  DATA 5,6,7,8,5
420  DATA 4,3,10,7,6,4
430  DATA 3,2,9,10,3
440  DATA 7,10,9,8,7
450  DATA 1,4,6,5,1
460  DATA 11,12,13,14,11
470  DATA 15,18,17,16,15
480  DATA 11,14,18,15,11
490  DATA 12,16,17,13,12
500  DATA 11,15,16,12,11
510  DATA 14,13,17,18,14
520  REM FILL SURFACE ARRAY
530  DIM S(13,6)
540  FOR I = 1 TO 13
550  FOR J = 1 TO NPS(I)
560  READ S(I,J)
570  NEXT J,I
```

表面陣列的執行結果將如下所示：

		J（被連接的點）					
	S(I,J)	1	2	3	4	5	6
	1	1	2	3	4	1	0
	2	1	5	8	9	2	1
	3	5	6	7	8	5	0
I	4	4	3	10	7	6	4
（表面）	5	3	2	9	10	3	0
	6	7	10	9	8	7	0
	7	1	4	6	5	1	0
	8	11	12	13	14	11	0
	9	15	18	17	16	15	0
	10	11	14	18	15	11	0
	11	12	16	17	13	12	0
	12	11	15	16	12	11	0
	13	14	13	17	18	14	0

法線陣列

法線陣列的計算和從前一樣，用N（ I ，1 ），N（ I ，2 ），和N（ I ，3 ）來存表面I 其方向朝外之法線向量的三個分量。

程式 7·4 ：第三部分

```
580  REM CALCULATE VECTOR NORMAL TO EACH SURFACE
590  DIM N(13,3)
600  FOR I = 1 TO 13
610 U1 = V(S(I,2),1) - V(S(I,1),1)
620 U2 = V(S(I,2),2) - V(S(I,1),2)
630 U3 = V(S(I,2),3) - V(S(I,1),3)
640 V1 = V(S(I,3),1) - V(S(I,1),1)
650 V2 = V(S(I,3),2) - V(S(I,1),2)
660 V3 = V(S(I,3),3) - V(S(I,1),3)
670 N(I,1) = U2 * V3 - V2 * U3
680 N(I,2) = U3 * V1 - V3 * U1
690 N(I,3) = U1 * V2 - V1 * U2
700  NEXT I
```

邊陣列

在程式 7.1 ，邊陣列提供的空間能存 12 個可見邊的端點。在此因多了煙囪，故再加上 12 個邊，但不論從任何角度觀測，此12 邊最多只有 9個爲可見。

因爲較遠物體的某些邊可能會被較近物體隱藏一部分，故參考“物體的編號”把每個邊分類將是很有用的。邊陣列被宣告爲E（ 2 ，12 ，3 ）。任一邊陣列E（ I ，J ，K ），I 的值究係 0 或 1 ，和此邊是物體 1 或物體 2（ 房屋或煙囪 ）有關。而對每個 I 而言，J 的值介於 1 和 12 之間，故允許每個物體可以有 12 個可見邊（ 房屋最多有 12 個可見邊，而煙囪最多有 9 個。）物體 I 上之邊 J ，E（ I ，J ，1 ）和E（ I ，J ，2 ）爲其邊之兩個端點。而E（ I ，J ，3 ）則是標記（ tag ），用來確認以此邊爲界限的可見表面之數目。

圖7.18 顯示了在邊陣列裏，邊的分類名稱。

I = 1；物體1（房屋）

邊	端點		標記
1	1	2	2
2	2	3	2
3	3	4	1
4	4	1	1
5	1	5	1
6	5	8	1
7	8	9	1
8	9	2	2
9	9	10	1
10	10	3	1
11	0	0	0
12	0	0	0

I = 2；物體2（煙囪）

邊	端點		標記
1	11	12	1
2	14	11	1
3	15	18	2
4	18	17	1
5	17	16	1
6	16	15	2
7	14	18	1
8	15	11	2
9	16	12	1
10	0	0	0
11	0	0	0
12	0	0	0

邊陣列的內容是由觀測點決定，若觀測點改變，邊的數目會改變，並且邊在邊陣列內的順序和標記也都會改變。

程式7·4：第四部分

```
710  REM DETERMINE VISIBILITY OF EACH SURFACE
720  REM AND COLLECT VISIBLE EDGES IN EDGE ARRAY
730 XE = RH * S2 * C1:YE = RH * S2 * S1:
    ZE = RH * C2
740  DIM E(2,12,3)
750 O = 1:N = 1
760  FOR I = 1 TO 13
770 E2 = S(I,1)
780 VX = XE - V(E2,1)
790 VY = YE - V(E2,2)
800 VZ = ZE - V(E2,3)
810  IF N(I,1) * VX + N(I,2) * VY + N(I,3) * VZ
     < = 0 THEN 920
820 E1 = S(I,1)
830  FOR J = 2 TO NPS(I)
```

```
840 E2 = S(I,J)
850  FOR K = 1 TO N
860  IF E(0,K,1) = E2 AND E(0,K,2) = E1
     THEN E(0,K,3) = 2: GOTO 900
870  NEXT K
880 E(0,N,1) = E1:E(0,N,2) = E2:E(0,N,3) = 1
890 N = N + 1
900 E1 = E2
910  NEXT J
920  IF I = 7 THEN 0 = 2:N = 1
930  NEXT I
```

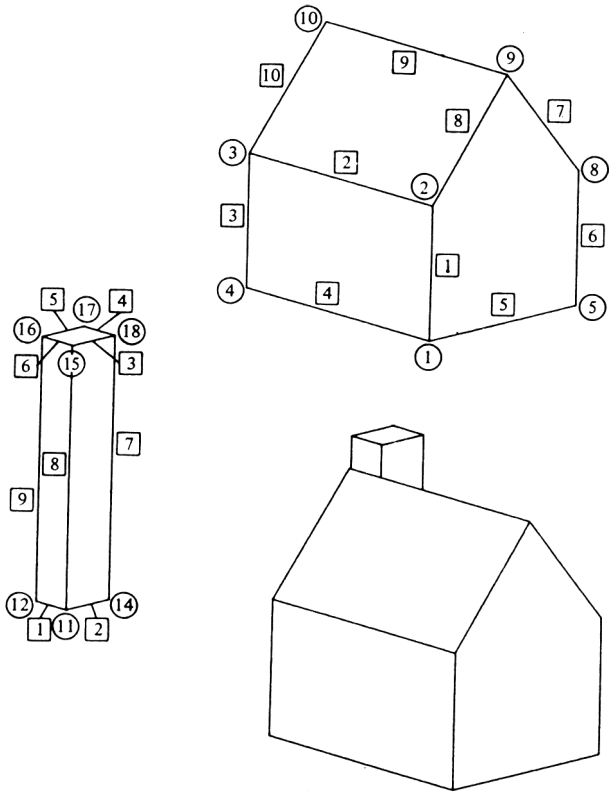


圖 7-18

這個程式對可見邊的確認，在本質上是和程式7.1一樣。物體1（房屋）的表面首先被考慮（行號750），而行號810測試其可見性，凡背對觀測點的，即被棄置。行號820到910接著考慮一個可見表面的每一邊。每一邊的端點由行號820到840所確認，並由行號880 滙集於邊陣列裏。而標記E（0，N，3）在行號880被設定爲1。若一個邊第二次被碰到（其即爲兩個可見表面的邊界。）則標記E（0，N，3）就被設定爲2（行號860）。當物體1（房屋）的所有表面都測試過可見性，則由物體2（煙囪）的表面繼續測試。

物體的優先順序（object priority）

因爲較遠物體的邊可能會被較近物隱藏邊的一部分，所以必須根據他們的相對位置把物體分類，並利用隔離面在兩個物體間建立其優先順序。平面Y＝－7即被用來做爲隔離面，因爲物體1的點有的在平面裏，有的在物體2的另一邊。

如果觀測點（XE，YE，ZE）和房屋在隔離面的同一邊，則房屋的可見邊不能被煙囪的部分所隱藏。故房屋爲第一優先，而煙囪爲第二優先，且其邊可能會部分或全部的被房屋所隱藏。若觀測點是和煙囪在隔離面的同一邊，則煙囪即爲第一優先，而房屋爲第二優先。

程式7·4：第五部分

```
940   REM ESTABLISH PRIORITY OF OBJECTS
950   IF YE > = -7 THEN P(1) = 1:P(2) = 2
960   IF YE < -7 THEN P(1) = 2:P(2) = 1
```

第一優先的邊

現在我們就可以把第一優先物體的可見邊畫出來。每個邊皆爲通過利用表面方位測試可見性的表面之邊界。因爲這些邊屬於第一優先物體，所以並不會被其他物體所隱藏（部分地或全部地）。

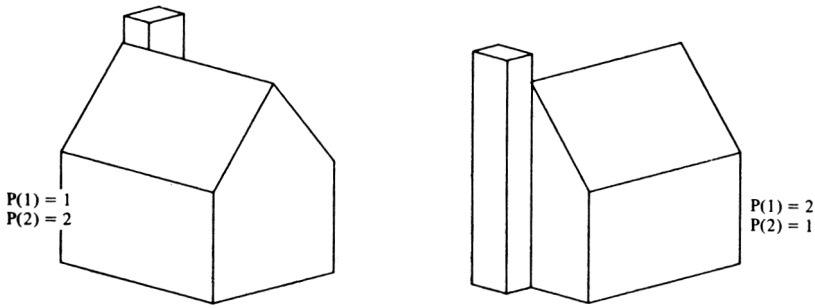


圖 7-19

程式 7·4 ：第六部分

```
970   REM DRAW VISIBLE EDGES OF FIRST PRIORITY
      OBJECT
980   HGR2 : HCOLOR= 3
990   FOR I = 1 TO 12
1000   IF E(P(1),I,3) = 0 THEN 1030
1010 J = E(P(1),I,1):K = E(P(1),I,2)
1020   HPLOT SV(J,1),SV(J,2) TO SV(K,1),SV(K,2)
1030   NEXT I
```

第二優先的邊

屬於第二優先物體的邊，我們應該小心，詳細的檢查。在邊陣列裏的邊，很可能會被第一優先物體隱藏部分或全部。

爲了說明起見，現在我們以圖 7.20 的房屋和煙囪爲例說明之。

煙囪的邊 1 和 2 是通過可見性測試之煙囪的邊界。但這些邊卻被房屋完全隱藏，故不能畫出來。而邊 3 ，4 ，5 和 6 則完全可見，應被畫出來。另外，邊 7 ，8 ，和 9 只有部分可見，這些邊的每一個，都必須再決定此邊從那一點起變成可見，然後把可見部分畫出來。

因此，在第二優先的物體，將有三種型式的邊。有些邊被較近物體完全隱藏，有些只有被隱藏一部分，其他的則是完全可見。

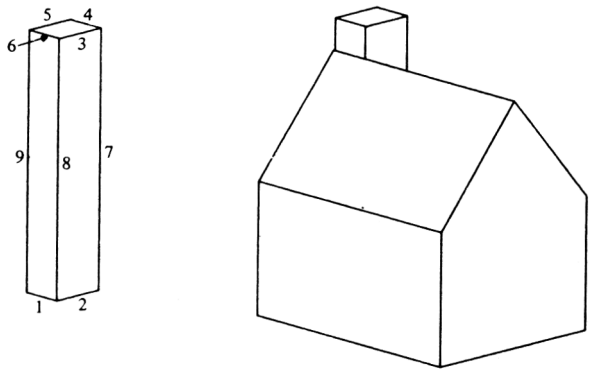


圖 7-20

程式 7·4 ：第七部分

```
1040  REM TEST EACH EDGE OF SECOND PRIORITY
      OBJECT FOR VISIBILITY STATUS
1050  REM DRAW VISIBLE EDGES AND VISIBLE PARTS
      OF EDGES
1060  FOR E = 1 TO 12
1070  IF E(P(2),E,3) = 0 THEN 1150
1080 I = E(P(2),E,1):J = E(P(2),E,2)
1090  IF SV(I,3) = 0 THEN XT = SV(I,1):
      YT = SV(I,2): GOSUB 1170:SV(I,3) = VI
1100  IF SV(J,3) = 0 THEN XT = SV(J,1):
      YT = SV(J,2): GOSUB 1170:SV(J,3) = VI
1110  IF SV(I,3) + SV(J,3) = -2 THEN 1140
1120  IF SV(I,3) + SV(J,3) = 0 THEN 1260
1130  HPLOT SV(I,1),SV(I,2) TO SV(J,1),SV(J,2)
1140  NEXT E
1150  END
```

程式的行號1040到1140 是用來監督第二優先各邊的配置情形。我們一直還未使用的顯示幕陣列 SV(I，3)，在此我們用來指出端點（由行號1080指定的）的可見性狀態（ status ）。行號1090和1100利用由行號1160 （底下將提到）開始的副程式來決定每個端點的可見性狀態。若一端點 I 爲可見，則 SV(I，3)就會被設定爲 1。若端點 I 爲隱藏，則 SV(I，3) 就被設定爲－ 1 。

一個由頂點 I 和 J 組成的邊，若 SV(I , 3) + SV(J , 3) = − 2 則此邊爲隱藏；且被棄置（ 行號 1110 ）。

若 SV(I , 3) + SV(J , 3) = 0（ 行號 1120 ），則此邊一端點爲可見，另一爲隱藏。而這些端點就整個被轉送到行號 1260 開始的副程式，以找出並畫出此邊的可見部分。

若 SV(I , 3) + SV(J , 3) = 2 ，則兩個端點皆爲可見；故此邊就被畫出來（ 行號 1130 ）。

邊上端點的可見性狀態

程式的行號 1080 到 1100 是確認每一個邊的端點，並將每一個端點由行號 1160 到 1250 分類成隱藏或可見。圖 7.21將被用來說明端點的分類。

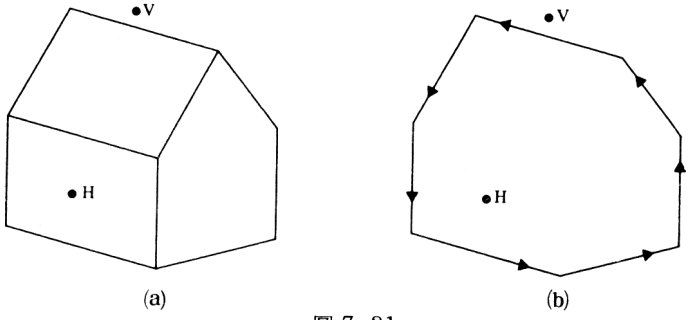


圖 7-21

在圖 7.21a，房屋的可見表面已被畫出來，並顯示出煙囪的兩個端點。注意應該被房屋隱藏的的端點 H ，是在房屋外部邊界的裏面（ 如圖 7.21b），而端點 V 則在外面。由於房屋的外部邊界爲逆時針方位（ 在表面陣列指定並保持到邊陣列），因此一個“內部”點 H ，可因爲它位於每一個邊界的左半平面而被確認。點 V 則可因其在至少一個邊界的右半平面而確認爲“外部”點。

程式 7·4 ：第八部分

```
1160  REM CLASSIFY ENDPOINTS OF EDGES AS VISIBLE
      OR HIDDEN
1170 VI = -1
1180  FOR L = 1 TO 12
1190  IF E(P(1),L,3) < > 1 THEN 1240
1200 P = E(P(1),L,1):Q = E(P(1),L,2)
1210 R1 = SV(Q,1) - SV(P,1):
     R2 = SV(Q,2) - SV(P,2)
1220 U1 = XT - SV(P,1):U2 = YT - SV(P,2)
1230  IF U2 * R1 - U1 * R2 > = 0
      THEN VI = 1: RETURN
1240  NEXT L
1250  RETURN
```

用來做爲分類的副程式，其主要目的在得到一個可見性的值，使VI 在可見端點（位於房屋邊界外面的點）時爲＋1，而在隱藏端點（位於房屋邊界裏面的點）時爲－1。欲達成此目的，我們可以先假設每個點都是隱藏的（行號1170：VI ＝－1），直到它被證明爲可見。

因爲一個點如果是位於至少一個邊界的右半平面，即爲可見，所以1180到1240的廻路把端點（XT，YT）跟每一個房屋的可見邊比較，但爲區別“內部”或“外部”，我們只考慮屬於外部邊界的可見邊。行號1190會跳過其他邊不管。

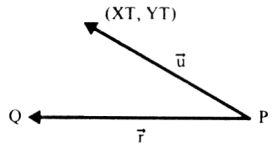


圖 7-22

對房屋的每一個外部邊PQ，有兩個向量要被計算。如圖7.22所示，向量$\vec{r}$從P延伸到Q，向量$\vec{u}$從P延伸到被分類的點（XT，YT）。如果（XT，YT）位於線段PQ 的右半平面（即可見），則u×r爲（0，0，1）的正倍數（請參閱附錄3）。在這種情況，程式的行號1230會把VI的值變爲1，然後結束副程式而回返。

若端點(XT,YT)是位於每一個外部邊界的左半平面，則VI = 1而從副程式回返主程式。

對部份隱藏的邊截割

若一個邊，有一端點爲隱藏，另一爲可見，則我們必須對這個線段截割以只顯示可見部分。程式的行號1120把屬於這類的邊轉送到行號1270開始的程式，在那裏，可見的點用(XV,YV)表示，而隱藏點則用(XH,YH)表示。

行號1290到1360使用"二分找尋法(binary search)"來找到點(XT,YT)，而把線段分成兩個部分：從(XV,YV)到(XT,YT)的可見部分和從(XT,YT)到(XH,YH)的隱藏部分。二分找尋法剛開始時，可能會讓我們覺得很慢且漫無目標，不過事實上它是很有效率的。其過程如下：

1. 使向量(V1,V2)等於從(XV,YV)延伸到(XH,YH)的一半。(行號1290)
2. 把(XV,YV)加上(V1,V2)而得到(XV,YV)和(XH,YH)之間的中點(XT,YT)。(行號1300)
3. 測試(XT,YT)的可見性。(行號1320)
4. 重新定義(V1,V2)=(V1,V2)/2。(行號1330)
5. 若(XT,YT)爲可見，則將其移向(XH,YH)，利用(XT,YT)=(XT,YT)+(V1,V2)。(行號1340，1350。)若(XT,YT)爲隱藏，則將其移向(XV,YV)，利用(XT,YT)=(XT,YT)-(V1,V2)。(行號1340，1350)
6. 回到第3步。(行號1360)

在測試(XT,YT)的可見性時(行號1320)使用到行號1170開始的分類副程式。如前所述，此副程式在返回主程式時，若(XT,YT)可見則VI =+1，若(XT,YT)隱藏則XI =-1。此時VI的值就可以決定下一個(XT,YT)的值是向(XH,YH)接近(若

VI ＝＋１），或向（ XV，YV ）接近（若VI ＝－１）。

從第３步到第５步的廻路，在計算完第７個（ XT，YT ）的值後即結束。此時（ V1，V2 ）的長度爲（ XV，YV ）和（ XH，YH ）之間距離的１／128 。這個解析度對大部分 APPLE 顯示幕上的圖形已經是非常足夠了。當然，只要增加廻路的週期也是可以得到更大的解析度。

在離開這個廻路時，從（ XV，YV ）到（ XT，YT ）的線段就已經被畫出來了。然後程式的控制就轉到行號 1070 ，而繼續判斷其他的邊。

程式 7·4 ：第九部分

```
1260  REM PLOT VISIBLE PORTION OF EDGES WHICH
      ARE PARTLY HIDDEN
1270  IF SV(I,3) = 1 THEN XV = SV(I,1):YV =
      SV(I,2):XH = SV(J,1): YH = SV(J,2)
1280  IF SV(J,3) = 1 THEN XV = SV(J,1):YV =
      SV(J,2):XH = SV(I,1): YH = SV(J,2)
1290 V1 = (XH - XV) / 2:V2 = (YH - YV) / 2
1300 XT = XV + V1:YT = YV + V2
1310  FOR I = 2 TO 7
1320  GOSUB 1170
1330 V1 = V1 / 2:V2 = V2 / 2
1340 XT = XT + VI * V1
1350 YT = YT + VI * V2
1360  NEXT I
1370  HPLOT XV,YV TO XT,YT
1380  NEXT E
```

程式 7·4 ：消除隱藏線（兩個物體；任意觀測點）

```
1  REM PROGRAM 7.4 (HIDDEN LINES)
2  REM TWO OBJECTS; ARBITRARY VIEWPOINT
10  REM VIEWPOINT AND SCREEN PARAMETERS
20 RHO = 40:THETA = .7:PHI = 1:D = 250
30 CX = 140:CY = 96:S1 = SIN (TH):C1 = COS (TH):
   S2 = SIN (PH):C2 = COS (PH)
40  REM VERTICES OF OBJECT 1 (HOUSE)
50  DATA 5,7,-5
```

```
60  DATA 5,7,5
70  DATA 5,-7,5
80  DATA 5,-7,-5
90  DATA -5,7,-5
100  DATA -5,-7,-5
110  DATA -5,-7,5
120  DATA -5,7,5
130  DATA 0,7,8
140  DATA 0,-7,8
150  REM VERTICES OF OBJECT 2 (CHIMNEY)
160  DATA -2,-7,-5
170  DATA -2,-9,-5
180  DATA -5,-9,-5
190  DATA -5,-7,-5
200  DATA -2,-7,9
210  DATA -2,-9,9
220  DATA -5,-9,9
230  DATA -5,-7,9
240  REM FILL VERTEX ARRAYS
250  DIM V(18,3),SV(18,3)
260  FOR I = 1 TO 18
270  READ X,Y,Z
280 V(I,1) = X:V(I,2) = Y:V(I,3) = Z
290 X1 = -X * S1 + Y * C1:Y1 = -X * C1 * C2 - Y *
    S1 * C2 + Z * S2:Z1 = -X * S2 * C1 - Y * S2 *
    S1 - Z * C2 + RHO
300 SV(I,1) = D * (X1 / Z1) + CX:
    SV(I,2) = -D * (Y1 / Z1) + CY
310  NEXT I
320  REM NUMBER OF POINTS IN EACH SURFACE
330  DATA 5,6,5,6,5,5,5,5,5,5,5,5,5
340  DIM NPS(13)
350  FOR I = 1 TO 13
360  READ NPS(I)
370  NEXT I
380  REM SURFACE ARRAY - POINTERS TO VERTICES
390  DATA 1,2,3,4,1
400  DATA 1,5,8,9,2,1
410  DATA 5,6,7,8,5
420  DATA 4,3,10,7,6,4
430  DATA 3,2,9,10,3
440  DATA 7,10,9,8,7
450  DATA 1,4,6,5,1
460  DATA 11,12,13,14,11
470  DATA 15,18,17,16,15
480  DATA 11,14,18,15,11
490  DATA 12,16,17,13,12
500  DATA 11,15,16,12,11
510  DATA 14,13,17,18,14
520  REM FILL SURFACE ARRAY
530  DIM S(13,6)
```

(continued)

```
540  FOR I = 1 TO 13
550  FOR J = 1 TO NPS(I)
560  READ S(I,J)
570  NEXT J,I
580  REM CALCULATE VECTOR NORMAL TO EACH SURFACE
590  DIM N(13,3)
600  FOR I = 1 TO 13
610 U1 = V(S(I,2),1) - V(S(I,1),1)
620 U2 = V(S(I,2),2) - V(S(I,1),2)
630 U3 = V(S(I,2),3) - V(S(I,1),3)
640 V1 = V(S(I,3),1) - V(S(I,1),1)
650 V2 = V(S(I,3),2) - V(S(I,1),2)
660 V3 = V(S(I,3),3) - V(S(I,1),3)
670 N(I,1) = U2 * V3 - V2 * U3
680 N(I,2) = U3 * V1 - V3 * U1
690 N(I,3) = U1 * V2 - V1 * U2
700  NEXT I
710  REM DETERMINE VISIBILITY OF EACH SURFACE
720  REM AND COLLECT VISIBLE EDGES IN EDGE ARRAY
730 XE = RH * S2 * C1:YE = RH * S2 * S1:
    ZE = RH * C2
740  DIM E(2,12,3)
750 O = 1:N = 1
760  FOR I = 1 TO 13
770 E2 = S(I,1)
780 VX = XE - V(E2,1)
790 VY = YE - V(E2,2)
800 VZ = ZE - V(E2,3)
810  IF N(I,1) * VX + N(I,2) * VY + N(I,3) * VZ
     < = 0 THEN 920
820 E1 = S(I,1)
830  FOR J = 2 TO NPS(I)
840 E2 = S(I,J)
850  FOR K = 1 TO N
860 IF E(O,K,1) = E2 AND E(O,K,2) = E1
    THEN E(O,K,3) = 2: GOTO 900
870  NEXT K
880 E(O,N,1) = E1:E(O,N,2) = E2:E(O,N,3) = 1
890 N = N + 1
900 E1 = E2
910  NEXT J
920  IF I = 7 THEN O = 2:N = 1
930  NEXT I
940  REM ESTABLISH PRIORITY OF OBJECTS
950  IF YE > = -7 THEN P(1) = 1:P(2) = 2
960  IF YE < -7 THEN P(1) = 2:P(2) = 1
970  REM DRAW VISIBLE EDGES OF FIRST PRIORITY
     OBJECT
980  HGR2 : HCOLOR= 3
990  FOR I = 1 TO 12
1000  IF E(P(1),I,3) = 0 THEN 1030
```

```
1010 J = E(P(1),I,1):K = E(P(1),I,2)
1020  HPLOT SV(J,1),SV(J,2) TO SV(K,1),SV(K,2)
1030  NEXT I
1040  REM TEST EACH EDGE OF SECOND PRIORITY
      OBJECT FOR VISIBILITY STATUS
1050  REM DRAW VISIBLE EDGES AND VISIBLE PARTS
      OF EDGES
1060  FOR E = 1 TO 12
1070  IF E(P(2),E,3) = 0 THEN 1150
1080 I = E(P(2),E,1):J = E(P(2),E,2)
1090  IF SV(I,3) = 0 THEN XT = SV(I,1):
      YT = SV(I,2): GOSUB 1170:SV(I,3) = VI
1100  IF SV(J,3) = 0 THEN XT = SV(J,1):
      YT = SV(J,2): GOSUB 1170:SV(J,3) = VI
1110  IF SV(I,3) + SV(J,3) = -2 THEN 1140
1120  IF SV(I,3) + SV(J,3) = 0 THEN 1260
1130  HPLOT SV(I,1),SV(I,2) TO SV(J,1),SV(J,2)
1140  NEXT E
1150  END
1160  REM CLASSIFY ENDPOINTS OF EDGES AS VISIBLE
      OR HIDDEN
1170 VI = -1
1180  FOR L = 1 TO 12
1190  IF E(P(1),L,3) < > 1 THEN 1240
1200 P = E(P(1),L,1):Q = E(P(1),L,2)
1210 R1 = SV(Q,1) - SV(P,1):
     R2 = SV(Q,2) - SV(P,2)
1220 U1 = XT - SV(P,1):U2 = YT - SV(P,2)
1230  IF U2 * R1 - U1 * R2 > = 0
      THEN VI = 1: RETURN
1240  NEXT L
1250  RETURN
1260  REM PLOT VISIBLE PORTION OF EDGES WHICH
      ARE PARTLY HIDDEN
1270  IF SV(I,3) = 1 THEN XV = SV(I,1):
      YV = SV(I,2):XH = SV(J,1): YH = SV(J,2
1280  IF SV(J,3) = 1 THEN XV = SV(J,1):
      YV = SV(J,2):XH = SV(I,1): YH = SV(I,2)
1290 V1 = (XH - XV) / 2:V2 = (YH - YV) / 2
1300 XT = XV + V1:YT = YV + V2
1310  FOR I = 2 TO 7
1320  GOSUB 1170
1330 V1 = V1 / 2:V2 = V2 / 2
1340 XT = XT + VI * V1
1350 YT = YT + VI * V2
1360  NEXT I
1370  HPLOT XV,YV TO XT,YT
1380  NEXT E
```

注意和建議

1. 這裏所使用的技巧並非沒有缺點，例如，如果第二優先物體其邊上之兩端點都在第一優先物體之外，則此邊就被畫出來。但這可能造成應該是部分隱藏的邊被畫出來。如圖 7.23 中，角柱體的一邊AB就將連被立方體隱藏的部分一起被畫出來。

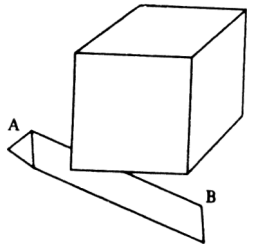


圖 7-23　立方體和角柱體

2. 修改程式的 DATA 和迴路的極限值，以畫出其他的圖形。例如圖 7.14 的角錐體和立方體。

3. 這裏使用的技巧也可擴展爲包含三個或更多的圖形。當然，程式的複雜性會增加，且執行時間也會增長。

4. 程式 7.4在圖形太大而超出顯示幕範圍時，並沒有提供線截割（line clipping）的處理，試修改程式使所有的線在被畫之前先經過截割副程式（如第四章所提到的）。

對Z＝F(X,Y) 形式的表面消除隱藏線

在第六章後面，我們列出一個程式做爲產生Z＝F（X，Y）圖型的例子。那個程式很簡要，也很快。不幸的是，它不能消除隱藏線，造成有時候圖形會變得混淆不清。在此，我們把它做一番修正，使

我們能得到像封面右上角所示的圖形。

在提到隱藏線的計算之前，我們再看一次以前的程式：

程式 7·5 ：表面

```
1  REM PROGRAM 7.5 (SURFACE)
2  REM REPEAT OF PROGRAM 6.4
10  REM INITIALIZATION
20  HGR2 : HCOLOR= 3:RHO = 30:THETA = .4:PHI =
    1.25: D = 420:CX = 140:CY = 96
30 S1 = SIN (THETA):C1 = COS (THETA)
40 S2 = SIN (PHI):C2 = COS (PHI)
50  DEF FN F(X) = COS (.1 * (X * X + Y * Y))
60  REM POINT GENERATING LOOP
70  FOR X = 10 TO -10 STEP -1
80 FL = 0
90  FOR Y = -10 TO 10 STEP .4
100 Z = FN F(X)
110  GOSUB 150
120  NEXT Y,X
130  END
140  REM PLOTTING SUBROUTINE
150 XE = -X * S1 + Y * C1
160 YE = -X * C1 * C2 - Y * S1 * C2 + Z * S2
170 ZE = -X * S2 * C1 - Y * S2 *
    S1 - Z * C2 + RHO
180 SX = D * (XE / ZE) + CX
190 SY = -D * (YE / ZE) + CY
200  IF SX < 0 OR SX > 279 OR SY < 0 OR SY > 191
     THEN FL = 0: RETURN
210  IF FL = 0 THEN HPLOT SX,SY:FL = 1
220 HPLOT TO SX,SY
230 RETURN
```

程式的行號 20 到 50是設定觀測助變數，顯示幕助變數，以及被畫的函數。而行號 70 到 120 的巢狀迴路（ nested loops ）是產生表面上點的座標，然後送到行號 150開始的畫圖副程式，在那裏，顯示幕座標（ SX，SY ）被計算出來（行號 150～190 ），若計算出來的座標值超過顯示幕，則將之棄置（行號 200 ）。每一條曲線的第一個點（ FL = 0 ），或在超出顯示幕的點之後的第一個在顯示幕內的點（同樣 FL = 0 ），則經由 HPLOT SX，SY 被畫出來（行號 210）

。否則，就連接最近被畫的點而畫出一線段（行號220），然後程式又回到產生點的迴路。

程式7.6把上面的程式擴充，而包括了隱藏線的能力。其起始設定（行號10～70）和程式7.5的開始很相似，設定觀測和顯示幕的助變數，以及被畫的函數。然後，行號60和70宣告兩個陣列，以供隱藏線程式之用，這個陣列的目的爲：

YX(XP)〔Y－maX〕確認對於每一個水平顯示幕座標XP，所遇見過的最大垂直顯示幕座標（最低的顯示幕位置）。YN(XP)〔Y－miN〕則確認對於每一個水平顯示幕座標XP，所遇見過的最小垂直顯示幕座標（最高的顯示幕位置）。

剛開始時，每一個YX(I)的值爲零，且每一個YN(I)爲191。當點（XP，YP）被畫出來後，YX(XP)和YN(XP)就被重新設定。當每個（XP，YP）被考慮畫出來時，先將YP和YX(XP)及YN(XP)比較，如果YP是位於YX(XP)和YN(XP)之間，則此點（XP，YP）就被分類成隱藏。否則就認定爲可見。如圖7.24所示。

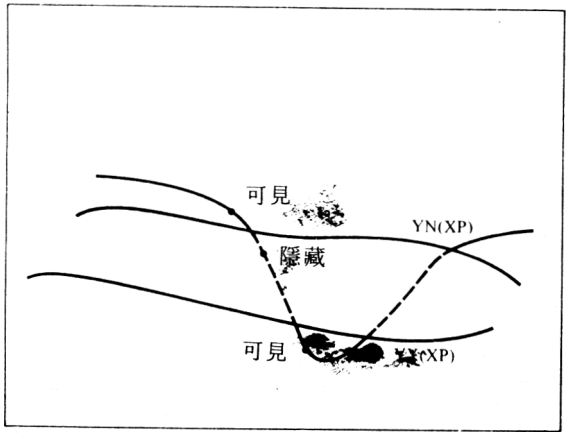
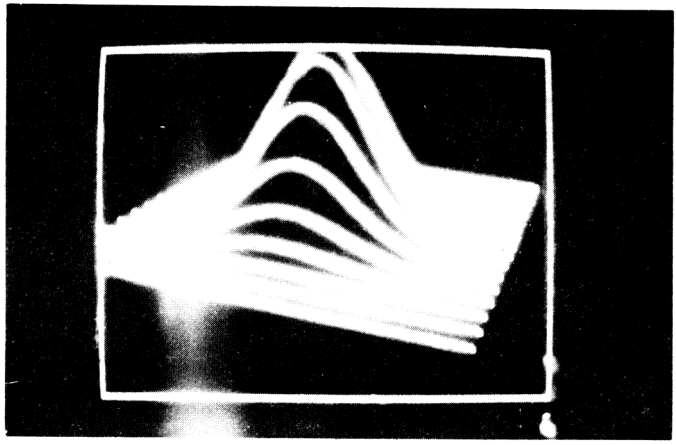


圖 7-24

程式 7·6：隱藏線（對 Z = F（X，Y）形式的表面）

```
1  REM PROGRAM 7.6 (HIDDEN LINES) .
2  REM HIDDEN LINE ELIMINATION FOR SURFACES: Z =
   F(X,Y)
10  REM INITIALIZATION
20  HGR2 : HCOLOR= 3:RHO = 30:THETA = .4:
    PHI = 1.25: D = 420:CX = 140:CY = 96
30 S1 = SIN (THETA):C1 = COS (THETA)
40 S2 = SIN (PHI):C2 = COS (PHI)
50  DEF FN Z(X) = 7 * EXP (-.1 *
    (X ^ 2 + Y ^ 2))
60  DIM YN(280),YX(280)
70  FOR I = 0 TO 279:YN(I) = 191: NEXT I
80  REM POINT GENERATING LOOP
90  FOR X = 8 TO -8 STEP -1
100 FL = 0
110  FOR  Y = -8 TO 8 STEP .5
120 Z = FN Z(X)
130  GOSUB 190
140  NEXT Y,X
150  HCOLOR= 2
160  HPLOT 0,0 TO 278,0 TO 278,191 TO 0,191 TO
     0,0
170  END
180  REM PLOTTING SUBROUTINE
190 XE = -X * S1 + Y * C1
200 YE = -X * C1 * C2 - Y * S1 * C2 + Z * S2
210 ZE = -X * S2 * C1 - Y * S2 * S1 - Z * C2 +
    RHO
220 SX = D * (XE / ZE) + CX
230 SY = -D * (YE / ZE) + CY
240  IF FL = 0 THEN FL = 1:F = 0: GOTO 360
250 DX = OX - SX: IF DX = 0 THEN DX = 1
260 SL = (OY - SY) / DX:YP = OY
270  FOR XP = INT (OX) + 1 TO SX
280 FG = 1
290 YP = YP + SL
300 IF XP < 0 OR XP > 279 THEN FG = 0:F = 0:
    GOTO 350
310  IF YP > 191 OR YP < 0 THEN FG = 0:F = 0
320  IF YP < = YN(XP) THEN 370
330  IF YP > = YX(XP) THEN 420
340 F = 0
350  NEXT XP
360 OX = SX:OY = SY: RETURN
370 YN(XP) = YP
380  IF FG = 0 THEN 410
390  IF F = 0 THEN HPLOT XP,YP:F = 1
400  HPLOT TO XP,YP
410  IF YP < YX(XP) THEN 350
```

```
420 YX(XP) = YP
430  IF FG = 0 THEN 460
440  IF F = 0 THEN HPLOT XP,YP:F = 1
450  HPLOT TO XP,YP
460  GOTO 350
```



使用這個陣列時，要求必須從一連串構成表面的曲線中，離觀測者最近的開始。產生點的迴路（行號 80 － 140 ）正是如此。

當每一個點（ X ，Y ，Z ）產生後，就被送到隱藏線副程式，而計算出其對應的顯示幕座標（ SX,SY ）。

然後點（ SX,SY ）就可以被測試其可見性（即判斷其究竟是 SY＜＝YN(SX)，或是SY＞＝YX(XP)？），但這樣還不夠，必須再決定每一點（ XP,YP ）在（ SX,SY）到前一個計算出的點（ OX,OY)（ Old point ）的線段上之可見性。如果（ XP,YP ）是隱藏，就不畫出來。如果（ XP,YP ）爲可見，則此點就被畫出來，並且其對應的陣列（ YX(XP)或YN(XP)）也被更新。

隱藏線副程式使用三個變數做爲旗號以告知動作狀況。

1. FL：如果點（SX，SY）是曲線的第一點，則FL = 0；否則FL = 1。
2. F：如果F = 0，則指令HPLOT XP，YP 就被用來畫出一個可見點（XP，YP），若F = 1，則使用HPLOT TO XP，YP這個指令。
3. FG：若現在的點（XP，YP）超出顯示幕，則FG = 0；否則FG = 1。

曲線上的第一點，我們利用設定FL = 0來做記號。（行號100）。在副程式裏，它被用來做爲一個判斷信號（行號240），以決定是否定義（OX，OY）（old point）爲最近被計算出的（SX，SY），並將程式返回產生點的廻路。

如果FL = 1，則線段（OX，OY）到（SX，SY）上的點就被測試可見性。行號250到260求出此線段的斜率（slope）。行號270開始一個整數XP（顯示幕座標）的廻路，其範圍從OX到SX。在YP被計算出後（行號290），即測試點（XP，YP）是否位於顯示幕內，若是，則將YP和陣列YN(XP)和YX(XP)的值比較（行號320，330和410）以決定點（XP，YP）的可見性。

如果YN(XP)＜YP＜YX(XP)，則點（XP，YP）爲不可見，且F被設定爲0（行號340），然後繼續考慮下一個點（行號350）。

如果YP＜＝YN(XP)，即YP不大於任一以前和XP有關的垂直顯示幕座標。故點（XP，YP）爲可見，位於以前被畫出來的表面部分之上。然後陣列YN(XP)的值就重設定爲YP（行號370），並畫出點（XP，YP）（行號390和400）。

如果YP＞＝YX(XP)（行號330和410）即YP比任一以前和XP有關的垂直顯示幕座標還大。則點（XP，YP）爲可見，並位於以前被畫出的表面部分之下。然後陣列YX(XP)的值就被重設定爲YP（行號420），並畫出點（XP，YP）（行號440和450）。

當所有從線段（OX，OY）到（SX，SY）的整數值XP 都被執行過，（OX，OY）就重新定義爲（SX，SY）的值，然後程式回返產生點的廻路（行號360）。

注意和建議

1. 程式7.6假設每一條曲線是以在顯示幕上從左到右的方向產生的。若要得到更具變通性的副程式可加上一行敍述如下：

```
265 ST = SGN(SX - OX)
```

並更改行號270爲

```
270 FOR XP = INT (OX) + ST to SX STEP ST
```

再把行號290改成

```
290 YP = YP + ST * SL
```

2. 有某些點（XP，YP）可能同時會被確認爲顯示幕最大值（YP>=YX(XP)）和顯示幕最小值（YP<=YN(XP)）。如此將造成YX(XP)和YN(XP) 同時都被重新設定，而此點（或線段）就被畫出兩次（行號390，400，440，和450 ），雖然這樣會浪費一些時間，不過倒也不至於造成損害。您是否能把它修正？

3. 在本章已經講了一些隱藏線的技巧，試利用它們來產生其他的圖形，如被土星環環繞的土星；或被一個洞穿過的立方體；或圖7.23的立方體和角柱體。

4. 在這章剛開始的敍述，我們曾講過以下這句話：

消除隱藏線和隱藏面是電腦繪圖最富挑戰性的問題之一。

的確，它是一個問題，我們也花了很多時間發展出一些隱藏線的技巧。不過在這裏都只是很簡要的介紹，如果讀者眞的有興趣從事這門學問，則應參考一些文章，如紐曼（Newman）和史普勞（Sproull）合著之“對談式電腦繪圖原理（Principles of Interactive Computer Graphics）”。

5. 本書前言曾提到的磁碟，也包含一些機器語言型式的求隱藏線

副程式。利用這些副程式將可以增加程式的執行速度。

第五部份

動　態

第八章　動態圖形

用HPLOT 畫動態圖形

在前面幾章，我們一直把焦點放在如何得到平面或立體的靜態圖形。因此，現在我們針對簡單的圖形提出一些“動態”的方法。當然，這些都和影片的原理一樣，我們只是顯示一連串被模擬移動的靜止圖形而已。前面我們也有幾個程式能產生動態圖形，本章將把這些方法加以詳細討論。

程式 1.2 就是一個模擬的跳球（ bouncing ball ）程式。首先在螢幕上畫出兩點以代表球，然後把它擦掉並再畫在另一新的位置即可。在這裏因爲圖形很簡單，只有兩個點，所以在一連串的 HPLOT 命令之間只需要很少的計算，因此球的移動看起來很平穩，逼眞。

程式 2.9也是一個類似的遊戲。不過這次因爲它的圖形比較複雜（一個方塊須要畫出四邊），當畫出圖形，擦掉，再畫出來時，若仔細看的話，可以發現其間的閃爍（ flicker ）現象。爲了避免這種閃爍現象，我們安排兩頁的高解像度圖形頁來一起使用。當其中一頁被用來顯示時，另外一頁就在做擦掉和畫的動作，如果畫好了就把它顯示出來，而剛才在顯示的那一頁此時就擦掉舊圖形，重新畫上新位置的圖形。如此循環不已，而構成一幅逼眞的動態圖形。

採用這種方式的程式必須執行下列的程序：

1. 控制用來被畫的頁數（ POKE 230,32 爲第一頁，POKE 230,64 爲第二頁）。
2. 控制被用來顯示的頁數（使用軟體開關來控制）。

3. 在每一連續圖形之間的計算過程須快而簡單。

第三個要求是很苛刻的，因爲大部分的動態圖形在每一連續圖形之間計算時間通常都很可觀。

> 注意：若要加速計算的時間，有一個方法就是利用機器語來寫程式。但如此一來設計程式的人可能需要耗費很大的精力，並且這部分本書也沒有提到。不過，最近APPLESOFT BASIC的編輯程式（compilers）已經變成可以很簡單的獲得一由機器語言轉換成的程式。如果您想使圖形更生動的話，您會發現編輯程式的重要性。

旋轉方塊 （Rotating Square）

如果產生一個連續圖形，而其中須要很繁瑣的運算時，或許我們可以事先把這些計算先算好，計算的結果留待以後繪圖時再拿出來使用。程式 8.1 就是利用這種方法。

首先把一個方塊的頂點都存在XO(I)和YO(I)這兩個陣列，而方塊的中心點在（0，0）。這個方塊我們並不準備拿來畫，而只是做爲推導出其他方塊座標的參考方塊。行號 180 到 150就是利用它來算出我們想要的 10 個方塊的頂點座標。每一個方塊都是把參考方塊旋轉一個角度 AN 並移到顯示幕裏面的位置後而獲得的（行號 90 和 110 ～ 130）。在第一個方塊，AN 的值爲 0，第二個則是 π / 20，第三個 2π / 20，餘依此類推（行號 20 和 90）。計算完的座標值都存在一組陣列X（I）和Y（I）裏。當 10 個方塊的座標都算好後，行號 160 到 180就把前兩個方塊的座標值再複寫一份，因此，第 11，12個方塊和第 1，2 個方塊是同一個。在此之所以多寫了兩個相同的方塊，乃是爲了配合程式消除圖像所設計的方法（即先消除圖

像，然後再畫出來）。

程式 8·1：旋轉方塊

```
1  REM PROGRAM 8.1 (ROTATING SQUARE)
2  REM ANIMATION VIA PRIOR CALCULATION
10  DIM X(60),Y(60)
20 PI = 3.14159:DT = PI / 20
30 C = COS (DT):S = SIN (DT)
40  DATA 80,0,0,80,-80,0,0,-80,80,0
50  FOR I = 1 TO 5
60  READ XO(I),YO(I)
70  NEXT I
80  FOR I = 0 TO 9
90 AN = I * DT
100 FOR J = 1 TO 5
110 T = COS (AN) * XO(J) - SIN (AN) *
    YO(J) + 140
120 Y(5 * I + J) = SIN (AN) * XO(J) + COS (AN) *
    YO(J) + 96
130 X(5 * I + J) = T
140  NEXT J
150  NEXT I
160  FOR I = 51 TO 60
170 X(I) = X(I - 50):Y(I) = Y(I - 50)
180  NEXT I
190  HGR : HGR2 :I = 1
200  HCOLOR= 0: POKE 230,32
210  HPLOT X(I),Y(I) TO X(I + 1),Y(I + 1)
     TO X(I + 2),Y(I + 2) TO X(I + 3),Y(I + 3)
     TO X(I + 4),Y(I + 4)
220 I = I + 10: HCOLOR= 3
230  HPLOT X(I),Y(I) TO X(I + 1),Y(I + 1)
     TO X(I + 2),Y(I + 2) TO X(I + 3),Y(I + 3)
     TO X(I + 4),Y(I + 4)
240  POKE - 16300,0: POKE 230,64:I = I - 5:
     HCOLOR= 0
250  HPLOT X(I),Y(I) TO X(I + 1),Y(I + 1)
     TO X(I + 2),Y(I + 2) TO X(I + 3),Y(I + 3)
     TO X(I + 4),Y(I + 4)
260 I = I + 10: HCOLOR= 3
270  HPLOT X(I),Y(I) TO X(I + 1),Y(I + 1)
     TO X(I + 2),Y(I + 2) TO X(I + 3),Y(I + 3)
     TO X(I + 4),Y(I + 4)
280  POKE - 16299,0:I= I - 5:IF I = 51 THEN I = 1
290  GOTO 200
```

注意：就理論上而言，從第一個方塊每次旋轉 π / 20，經過10次的旋轉後，方塊應該旋轉90°，變成和第一個方塊相同。然而事實上因爲在我們計算過程中的π值只是一個近似值，因此經過十次的旋轉必會累積一個誤差值。爲便於觀察，試將程式做如下之調整：

1. 行號80重寫如下：
   80FOR I = 0 TO 11
2. 去掉行號160，170和180
3. 重新執行程式

從行號200開始到290是控制方塊連續的顯示在顯示幕上。首先行號200到230先把第一頁圖形擦掉，然後再畫上去一個方塊，接下來行號240則把第一頁顯示出來，並將第二頁拿出來準備使用。行號250到270 同樣地先把第二頁擦掉再畫上去一個新的方塊，然後行號280把第二頁顯示出來，接著再繼續重複執行（行號290），直到最後一個方塊（第12個）畫出來以後，我們就再回到第一個，一直繼續畫下去（行號280）。

注意：在產生這些動態的圖形時，我們把圖形擦掉的方法是把它塗上和背景一樣的顏色——黑色。當然，我們也可以把顯示幕清除（CALL 62450）但這種方式較不能讓人滿意，除非是很複雜的圖形。因爲它花的時間比較長，同時還有個缺點：它把整個顯示幕，包括一些固定而不欲產生動態的圖形，都給清除掉了。

動態的立體圖形 （Three-Dimensional Animation）

接下來這個程式比上一個程式要長很多，不過基本上一樣是處理動態圖形。程式 9.2能顯示一個旋轉角錐。開始時角錐體的底在X Y平面，而頂點在Z 軸，然後這個角錐體就在這三個軸之間旋轉。

由於角錐在旋轉時會呈現出不同的邊，因此我們必須使用一些方法將不顯示的邊隱藏起來以獲得令人滿意的圖形。從第七章我們知道這種處理過程是很耗時的，而無法產生逼眞的動態圖形。因此，爲了讓圖形能以適當的速度顯示出來，我們必須把這些計算事先做好。

由於只要顯示一個圖形，我們可以使用測定表面方位法（ surface orientation method ）來決定顯示的表面。程式 8.2即採用此法，它包括了程式 7.1的大部分，而不同處底下我們將會指出來。

行號 10宣告一個整數陣列 SP％（I），這個陣列是用來儲存被計算好的角錐體可見邊的端點座標。角錐體最多可顯示 6 個邊，每一個邊有兩個端點，每個端點有 X 軸和 Y 軸這兩個座標值，因此總共有 24 個座標值。另外我們打算利用 36個不同角度的角錐圖形來達成旋轉的動作，這樣陣列的長度就有了 864（ 36 × 24）個。同時我們希望多出兩個與最初兩個圖形相同的圖形（其原因和上節所述相同），所以陣列就必須再提供第 37 ， 38 個圖，亦即陣列的長度將爲 912。

行號 40 和 50用來決定自始至終角錐體將被旋轉的角度，並把這些 SIN 和 COS 值存起來以爲程式做旋轉變換（ rotation transformation ）之用。

就像剛才提到的，我們將產生 36 個角錐圖形，這些連續圖形就在行號 260 到 740被計算和檢查圖形可見的部分，當發現一個可見邊時就把它的端點座標存在陣列 SP％（I）（行號 610），當一個角錐體的所有可見部分都已計算完畢，則定義角錐體的座標就會經由旋轉變換程式（行號 640 到 720 ）被變換。

行號 730 把現在的 RO 值（計算完的角錐體）印在顯示幕上，所以您可以知道這些計算如何被進行。

若 36 個圖形都已完成，行號 750 就把最前面兩個圖形複寫在第

37 ，38 個，以易於控制消除圖形。

至於把角錐體顯示出來的方法和程式 8.1 很類似。先從陣列SP%(I)裏拿出每個邊的座標值（行號 830 和 900），並且在行號 870 到 920 中把邊畫出來，而在行號 800 到 850 把它擦掉。同時 ADDR的值（用來做爲陣列的指標）在每個邊畫出後將會被調整（行號 840 和 910 ），或是在畫出每個角錐後（行號 860 和 930 ）。

> 注意：經過連續變換 36 次之後，我們怎麼知道角錐是否會回到原來位置呢？或許可能不會，不過在程式執行之初，會把角錐體的所有座標印出並比較，因此，我們可知角錐體經 36 次旋轉之後，幾乎回到其起始位置。

程式 8·2 ：旋轉角錐

```
1   REM PROGRAM 8.2 (ROTATING PYRAMID)
2   REM THREE DIMENSIONAL ANIMATION WITH HIDDEN
    LINES ELIMINATED
10   DIM SP%(912):ADDR = 1: DIM E(6,3)
20 RHO = 15:THETA = .5:PHI = .9:D = 400
30 CX = 140:CY = 96:S1 = SIN (TH):
    C1 = COS (TH):S2 = SIN (PH):C2 = COS (PH)
40 TN = -.1:TT = .1:CT = COS (TT):
    ST = SIN (TT):SO = SIN (TN):CO = COS (TN)
50 TP = -.1:SP = SIN (TP):CP = COS (TP)
60   DATA 0,0,3
70   DATA 1,0,0
80   DATA -.2,1,0
90   DATA -.2,-1,0
100   DIM V(4,3),SV(4,2)
110   FOR I = 1 TO 4: READ X,Y,Z
120 V(I,1) = X:V(I,2) = Y:V(I,3) = Z
130 XE   = -X * S1 + Y * C1:YE = -X * C1 *
    C2 - Y * S1 * C2 + Z * S2:ZE = -X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
140 SV(I,1) = D * (XE / ZE) + CX:
    SV(I,2) = -D * (YE / ZE) + CY
150   NEXT I
160   DATA 1,4,2,1
```

```
170  DATA 1,2,3,1
180  DATA 1,3,4,1
190  DATA 2,4,3,2
200  DIM S(4,4)
210  FOR I = 1 TO 4
220  FOR J = 1 TO 4
230  READ S(I,J)
240  NEXT J,I
250  DIM N(4,3)
260  FOR RO = 1 TO 36
270  FOR I = 1 TO 6:E(I,3) = 0: NEXT I
280  FOR I = 1 TO 4
290 U1 = V(S(I,2),1) - V(S(I,1),1)
300 U2 = V(S(I,2),2) - V(S(I,1),2)
310 U3 = V(S(I,2),3) - V(S(I,1),3)
320 V1 = V(S(I,3),1) - V(S(I,1),1)
330 V2 = V(S(I,3),2) - V(S(I,1),2)
340 V3 = V(S(I,3),3) - V(S(I,1),3)
350 N(I,1) = U2 * V3 - V2 * U3
360 N(I,2) = U3 * V1 - V3 * U1
370 N(I,3) = U1 * V2 - V1 * U2
380  NEXT I
390 XE = RH * S2 * C1:YE = RH * S2 * S1:
    ZE = RH * C2
400 N = 1
410  FOR I = 1 TO 4
420 E2 = S(I,1)
430 WX = XE - V(E2,1)
440 WY = YE - V(E2,2)
450 WZ = ZE - V(E2,3)
460  IF N(I,1) * WX + N(I,2) * WY + N(I,3) * WZ
     < = 0 THEN 570
470 E1 = S(I,1)
480  FOR J = 2 TO 4
490 E2 = S(I,J)
500  FOR K = 1 TO N
510  IF E(K,1) = E2 AND E(K,2) = E1
     THEN E(K,3) = 2: GOTO 550
520  NEXT K
530 E(N,1) = E1:E(N,2) = E2:E(N,3) = 1
540 N = N + 1
550 E1 = E2
560  NEXT J
570  NEXT I
580  FOR I = 1 TO 6
590  IF E(I,3) = 0 THEN 620
600 J = E(I,1):K = E(I,2)
610 SP%(AD) = SV(J,1):SP%(AD + 1) =
    SV(J,2):SP%(ADDR + 2) = SV(K,1):
    SP%(ADDR + 3) = SV(K,2)
620 ADDR = ADDR + 4
```

(continued)

```
630  NEXT I
640  FOR I = 1 TO 4
650 T1 = CP * CT * V(I,1) - (ST * CP + SO * SP) *
    V(I,2) + (SO * ST * CP - SP * CO) * V(I,3)
660 T2 = ST * V(I,1) + CO * CT * V(I,2) - SO *
    C: * V(I,3)
670 T3 = SP * CT * V(I,1) + (SO * CP - CO * ST *
    SP) * V(I,2) + (ST * SO * SP + CO * CP) *
    V(I,3)
680 V(I,1) = T1:V(I,2) = T2:V(I,3) = T3
690 X = T1:Y = T2:Z = T3
700 XE = -X * S1 + Y * C1:YE = -X * C1 *
    C2 - Y * S1 * C2 + Z * S2:ZE = -X * S2 *
    C1 - Y * S2 * S1 - Z * C2 + RHO
710 SV(I,1) = D * (XE / ZE) + CX:
    SV(I,2) = -D * (YE / ZE) + CY
720  NEXT I
730  PRINT RO
740  NEXT RO
750  FOR I = 1 TO 48:SP%(I + 864) =
     SP%(I): NEXT I
760  HOME : INPUT "READY";A$
770  HGR : HGR2 :DP = 0
780 ADDR = 1
790  POKE 230,32 * (DP + 1)
800  HCOLOR= 4
810  FOR I = 1 TO 6
820  IF SP%(ADDR) = 0 THEN 840
830  HPLOT SP%(ADDR),SP%(ADDR + 1)
     TO SP%(ADDR + 2),SP%(ADDR+3)
840 ADDR = ADDR+4
850  NEXT I
860 ADDR = ADDR+24
870  HCOLOR = 7
880  FOR I = 1 TO 6
890  IF SP%(ADDR) = 0 THEN 910
900  HPLOT SP%(ADDR),SP%(ADDR + 1)
     TO SP%(ADDR + 2), SP%(ADDR + 3)
910 ADDR = ADDR + 4
920  NEXT I
930 ADDR = ADDR - 48
940  IF ADDR = 865 THEN ADDR = 1
950  POKE - 16300 + DP,0
960 DP = 1 - DP
970  GOTO 790
```

造型表的動態圖形　（Shape Table Animation）

在使用HPLOT指令畫動態圖形時，通常我們都需要一些事先的準備工作，以使在顯示每一個連續圖形之間的計算過程減到最少。同樣地，若我們用造型表（shape table）的方法來達成模擬物體的移動時，因爲必須要準備一份供繪圖之用的表（table），所以它仍然需要一些事先的準備工作。

程式1.9（美術家的助手）就是利用造型表來做成一個可變長度的旋轉遊標。利用XDRAW和DRAW這兩個命令，配合顏色的使用，就可以使遊標在顯示幕上。

另一個例子則在附錄2的程式A2.1，用以模擬一個雷達顯示幕，使顯示幕出現一個旋轉的遊標和飛機。

在這幾個例子裏，都是利用在一連串的連續位置中做“畫”和“再畫（但顏色與背景同）”的動作來達成動態，程式8.3把這種程序加以擴充而模擬物體的移動。

程式8·3：尺蠖（inchworm）

```
1   REM PROGRAM 8.3 (INCHWORM)
2   REM SHAPE TABLE ANIMATION
10   DATA 5,0,12,0,21,0,30,0,39,0,46,0
20   DATA 45,40,32,5,168,174,21,37,0
30   DATA 45,40,32,5,40,45,21,37,0
40   DATA 45,5,40,5,40,45,21,37,0
50   DATA 45,45,45,45,45,37,0
60   DATA 9,5,32,5,40,168,21,21,173,37,0
70   FOR I = 768 TO 824
80   READ X: POKE I,X
90   NEXT I
100   POKE 232,0: POKE 233,3
110   HOME : HGR : HCOLOR= 3: SCALE= 1: ROT= 0
120   FOR X = 10 TO 100 STEP 4
130   FOR I = 1 TO 5
140   XDRAW I AT X,100
150   FOR T = 1 TO 60: NEXT T
160   XDRAW I AT X,100
170   NEXT I,X
```

造型表的圖形如圖8.1，此乃用來表示一隻蠕行尺蠖的位置。這

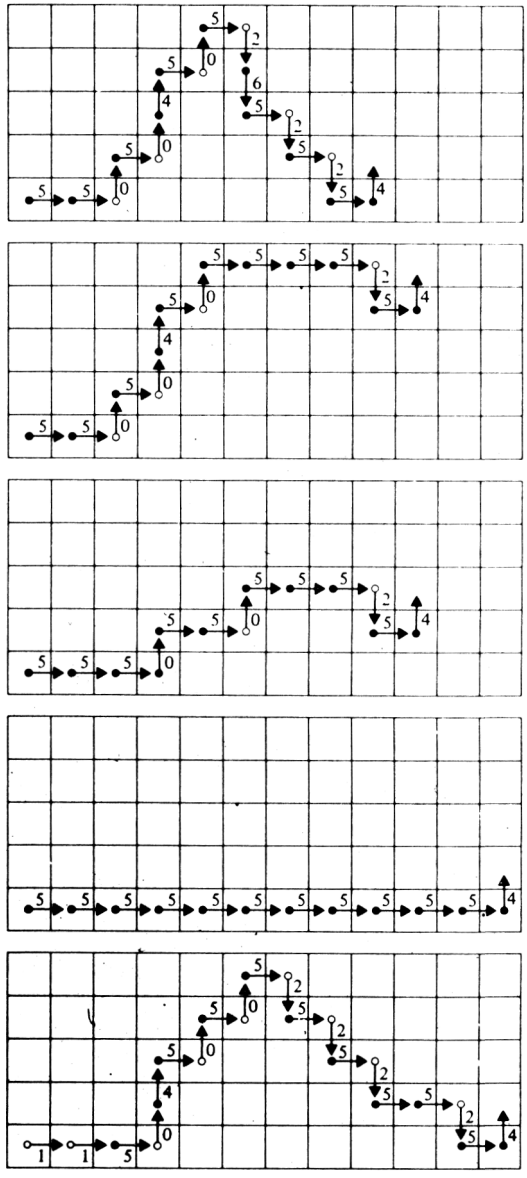


圖 8-1

個圖形可以利用程式A2.2（圖形的組合）來完成設計。當造型完成後就可以把它的內容放在DATA 敍述之後，並把它讀進程式同時指定其存放的位址（行號10－100）。而在程式的行號110到170我們利用XDRAW這個指令來做一連串畫圖，消除的動作，使五個圖形沿著顯示幕的水平線前進。

XDRAW這是一個用來顯示圖形用的指令，因爲它畫圖的方式是採用將目前顯示圖形的顏色取補色（complement）再畫出來的方式，所以要消除圖形就特別簡單。因爲XDRAW N AT X，Y 這個敍述會以顯示圖形目前顏色的補色畫出圖形N，而第二個XDRAW N AT X，Y 再取一次補色，結果就變成原來的顏色。這種特性將允許一個圖形通過另一個圖形而不致使圖形分裂，瓦解。

在程式8.3中，每一個圖形都是經由行號140而顯示來的，並在行號160中把它消除。一個延遲廻路（行號150）將可使圖形在消失前維持一段時間顯示在顯示幕上。

注意和建議

1. 對一簡單的圖形而言，畫和消除都可以很快完成，但是在畫比較複雜的圖形時就可能造成視覺上分散的感覺。此時我們就必須使用兩個圖形頁經由軟性開關（soft switch），使之交互顯示。

2. 利用HPLOT顯示一個旋轉的矢狀物（如時鐘的指針）的動態圖形。再試試看用造型表來達成。

3. 程式8.3是一個簡單的尺蠖圖形，試寫一個程式利用造型表以顯示一個人在走路。只要事先設計好幾個不同位置的手和脚的圖形即可達成。不過利用磁碟BLOAD進來的造型表資料（可由程式A2.2建立）要比在程式裏定義來得有效的多了。

4. 用同樣的方法寫一個程式顯示一個人在步上階梯的情形。

5. 在3和4所建議的程式中，試在顯示動態圖形前增添一些背景的景物。

6. 前面曾提到，若用機器語言的程式將會使動態圖形更快更平穩。如果您對BASIC 編輯程式的存取已有所了解，可以試著把前面幾個程式編輯過的型式看看。

7. 目前市面上已有一些簡單動態圖形的軟體程式，且都是用機器語言寫的，您也可以自己再加上一些資料以顯示其他的圖形。這些程式的模擬移動有些是用前面提過的用HPLOT配合兩個圖形頁和軟性開關的方法。其他產生的圖形則用第三章的點圖形文字（Bit Pattern Text）法來達成。

第六部份

週邊設備和軟體

第九章　系統的構成要件

一完整的電腦繪圖系統，須具備下列三種要件：

1. 輸入裝置：用來傳送資料和指令給電腦的裝置。
2. 處理單元：即電腦。
3. 輸出裝置：用以顯示圖像的裝置。

到目前爲止，本書所用到的輸入裝置就只有鍵盤，輸出裝置就只有電視螢幕。當然，其他還有很多種輸入／輸出裝置，底下我們將簡要的介紹幾種輸入／輸出裝置。

輸入裝置的硬體和軟體（Input Hardware And Software）

遊戲搖桿和按鈕（game paddles and pushbuttons）

由於遊戲搖桿在電視遊樂器（video games）方面的使用，微電腦也常採用它來做爲位移的測量。通常每一組遊戲搖桿有兩個輸出通道，以表示水平和垂直方向的位置。若用程式來控制每一個輸出通道，則其將提供一個0到255的整數給程式。

APPLE Ⅱ一共可接四個遊戲搖桿的輸出通道（亦即兩組遊戲搖桿）。我們可以利用PDL(I)（I＝0，1，2，或3）這個指令獲得其位置的值（0～255之間）。

程式9.1可將PDL(0)和PDL(1)的值顯示在螢幕。

程式9·1：搖桿的讀取

```
1   REM PROGRAM 9.1 (PADDLE READER)
2   REM DISPLAYS VALUES PROVIDED BY GAME PADDLES
10   TEXT : HOME
20   VTAB 1: PRINT PDL (0), PDL (1)
30   GOTO 20
```

請試試這個程式並搖動遊戲搖桿，觀察螢幕有何變化？

遊戲搖桿所提供的數字（ 0～255 之間）也可以用來做爲繪圖的應用。程式 9.2就是利用它來畫一圖案。我們可以發現經由搖桿操縱來繪圖就如同在螢幕上刻蝕一個圖案。（ 注意：行號 30 的敍述係爲了配合程式 9.3 在此並無意義。）

程式 9·2 ：電子筆

```
1   REM PROGRAM 9.2 (ELECTRONIC PEN)
2   REM USES PADDLES TO CONTROL DRAWING
10   HGR2 : HCOLOR= 3
20 X = 279 * PDL (0) / 255
30 Y = 191 * PDL (1) / 255
40   IF F = 0 THEN F = 1: HPLOT X,Y
50   HPLOT TO X,Y
60   GOTO 20
```

通常標準的遊戲搖桿都附有兩個按鈕（ pushbuttons），不過在 APPLE Ⅱ 上，最多可以接三個按鈕。每一個按鈕都可以用來發出開或關（ ON / OFF ）的信號，而這些信號可以從記憶體位址的－16287 ，－16286 ，和－16285接收到。正常時，記憶體位址裏的值小於 128（ 卽OFF ）。當按鈕被按時，則此按鈕所對應的記憶體位址，其值將大於或等於128（ 卽ON ）。

程式 9.3 將程式 9.2 的功能加强，使程式也能經由按鈕來繪圖。當按“按鈕 0”的時候，則程式將暫停繪圖，搖桿可以任意移動，並且它目前的位置也會被指示在螢幕上，但其移動軌跡並不會被畫出來。若按“按鈕 1”則恢復繪圖的功能。

程式 9·3：改良的電子筆

```
1   REM PROGRAM 9.3 (IMPROVED ELECTRONIC PEN)
2   REM INCLUDES PUSHBUTTON CONTROL
10   HGR2 : HCOLOR= 3
20 X = 279 * PDL (0) / 255
30 Y = 191 * PDL (1) / 255
40   IF F = 0 THEN F = 1: HPLOT X,Y
50   HPLOT   TO X,Y
60   IF PEEK ( - 16287) > 127 THEN 80
70   GOTO 20
80 X = 279 * PDL (0) / 255
90 Y = 191 * PDL (1) / 255
100   HCOLOR= 3: HPLOT X,Y
110   FOR I = 1 TO 10: NEXT I
120   IF PEEK ( - 16286) > 127
      THEN F = 0: GOTO 20
130   HCOLOR= 0: HPLOT X,Y
140   GOTO 80
```

光筆（light pens）

光筆的外形和鉛筆差不多，若將光筆的纜線和電腦之間連接妥當，則當其筆尖觸壓在螢光幕上時，其觸壓的位置將會被傳送到電腦，根據這些位置的指示，我們可以用來畫線，就如同前面提到的遊戲搖桿一樣。也可以用來指出在顯示幕上的圖像，在這種用法時，我們可以藉此指出圖像欲修改或更進一步研究的部分。另外也可以做一些如在顯示幕所列出菜單中選取菜餚之類的事。

繪圖板（graphics tablets）

繪圖板的功能有點像光筆，它也有一隻和電腦連接的“筆”，但並不是在電視螢幕畫圖，而是將“筆”觸壓在一塊也是連接到電腦的長方形薄板（tablets）上，而板子被觸壓的位置，也會被傳送到電腦去。

和光筆比起來，繪圖板或許有些不方便，因爲它是個獨立的裝置

，繪圖板的“筆”畫在板子上的圖形是顯示在電視螢幕而不是板子上，因此繪圖者就必須根據板子和螢幕的對應位置而在空無一物的板子上完成一幅圖案。

但繪圖板的優點也就在於它是個獨立的裝置，因爲板子是平放的，在平放的板子上畫圖是要比在豎立的電視螢幕上畫圖來得容易。此外，若將一張地圖覆於板子上，而把“筆”沿著地圖上的河流，公路，或地形特徵等來畫，那我們就可以把一張地圖複印在電視螢幕上，並且圖形的資料也存在電腦的記憶體裏，以後可以再拿出來使用。

典型的繪圖板其解像度（ resolution ）一般都比光筆要好，光筆在 APPLE Ⅱ電視螢幕的解像度若是 280 × 192 時繪圖板則遠超過此值，當圖像欲被放大，或圖像元素用於計算時，則這個增加的精確度就非常有價值。另外，如果顯示幕的解像度不敷使用時，可以換用其他的輸出設備。

圖 9.1 所示即爲一可在微電腦使用的繪圖板。圖 9.4則是一幅利用繪圖板所畫出來的圖形。

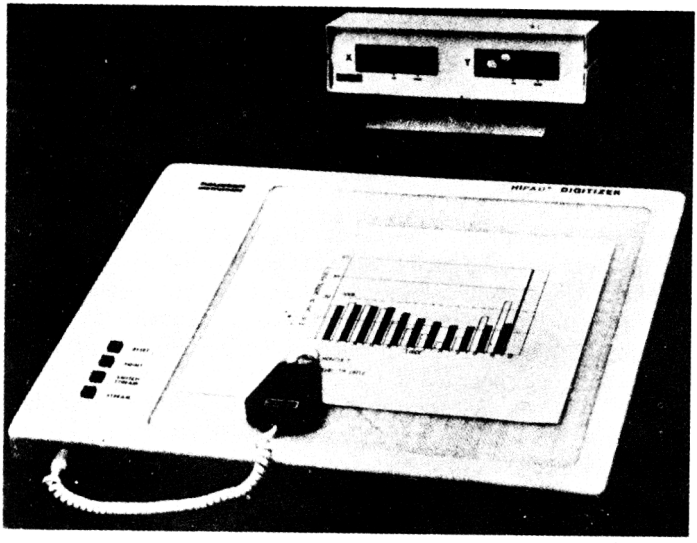


圖 9-1 繪圖板

目前已經有一些功能很强的軟體可供光筆和繪圖板使用。有了適當的軟體配合，“筆”要畫或抹除圖形都很簡單，並且圖形也可以放大，縮小，或旋轉，每一個圖形元素（element）皆可分別著色，甚至算出所繪物體的周長及面積。

輸出裝置的硬體和軟體（Output Hardware And Software）

繪圖機（plotters）

前面所提到的繪圖裝置都是輸入用的，在此要介紹的繪圖機則是供輸出圖像用的。它是用筆和墨水在紙上繪出圖像，至於筆繪圖的位置則由電腦控制，這和BASIC在螢幕上畫圖的指令HPLOT，和HPLOT TO 的指令有點類似。當然，當繪圖機用來幫助或代替電視螢幕時，送到繪圖機的資料必須符合它可接受的格式。

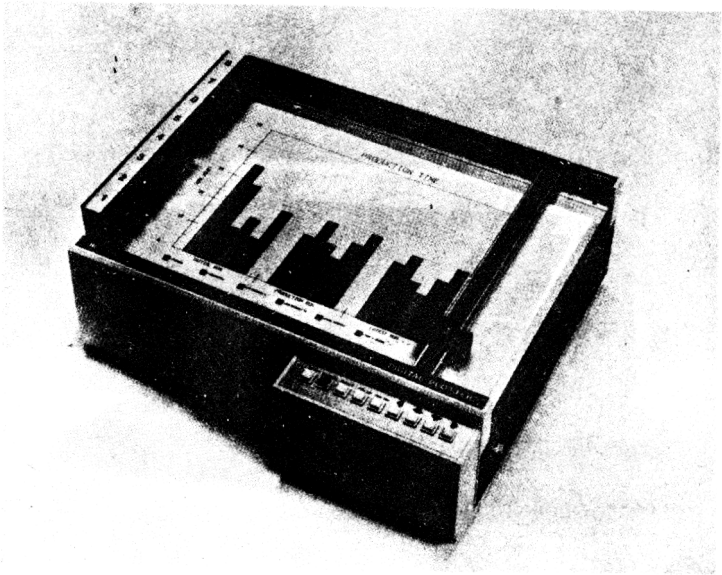


圖9-2　數位繪圖機

最基本的格式，就是將資料送到繪圖機定義其筆的狀態（上或下）和位置（希望筆到達的座標）。指令的順序如下：

```
UP
100,120
DOWN
UP
```

這將使筆提起，並移到（100,120）的位置，再將筆壓在紙上，再提起。因此就可以在紙上畫一個點。

若要畫一條從（100,200）到（200,10）的線，則其順序如下：

```
UP              （提筆）
100,120         （移到（100,120））
DOWN            （壓筆）
200,10          （移到（200,10））
UP              （提筆）
```

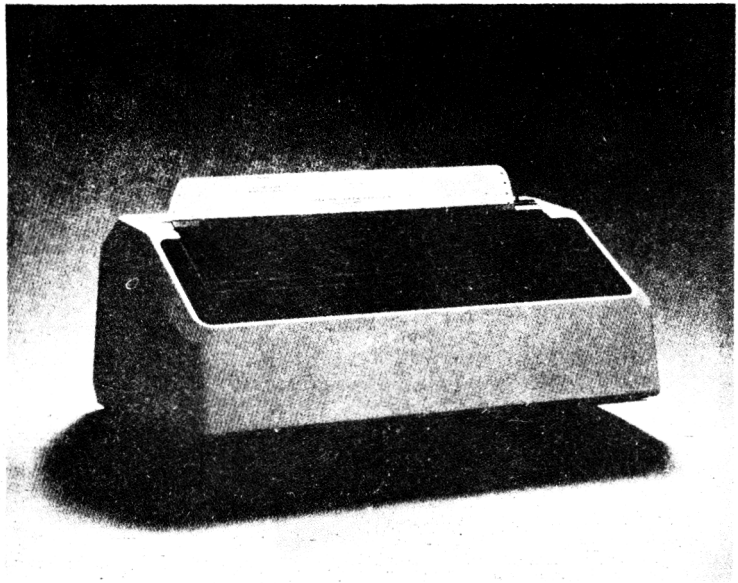


圖 9-3 點矩陣列表機

另外還可以利用內建的字型集（ built-in character sets ）而很容易的就能把一些文字，符號加在圖像上，線（ line ）的型式也可以有很多種變化（如實線，虛線，點線，點虛線等）另外可受控制的暫停（ pause）則能使筆停下來更換顏色筆，不過比較複雜的繪圖機則能經由指令的控制而命令其自動更換顏色筆。

利用繪圖機所畫出來的圖形在本書中的圖 4.12 ，5.1 ，和 7.1。

圖形列表機（graphics printers）

有一些列表機可以用來印出圖像。如點矩陣列表機（ dot matrix printer ，如圖 9.3），它的印字頭有一排打點，利用軟體程式的控制可以把存在記憶體繪圖頁裏的點圖形（ bit patterns ）資料以一個位元組接一個位元組（ byte-by-byte）的方式送到列表機，若位元組中某位元（ bit ）被設定爲ON ，則此位元所對應的印字頭打點就會打在紙上形成一點。一般而言，在同一時間內會有很多點圖形的位元組同時被送到列表機。另外還有一種熱感應列表機（ thermal

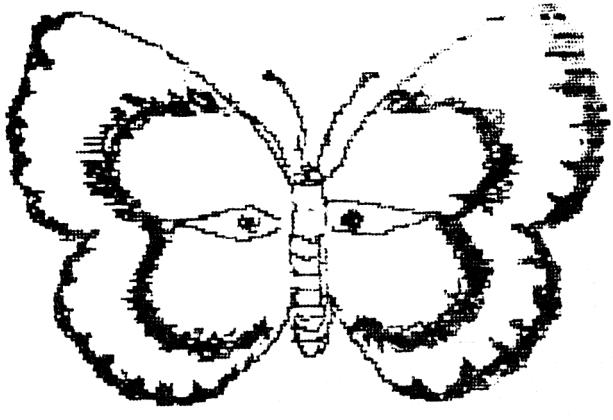


圖 9-4

printers ），它的動作方式和前述的點矩陣列表機很類似，只是它並非用墨水點在紙上，而是用熱來感應到紙上，形成圖形。

圖 9.4即利用點矩陣列表機所印出來的，而圖案就是利用前面所提到的繪圖板畫出來的。

附　錄

附錄 1　二進制，十進制與十六進制

人類會使用十進制據說也是一種巧合。假若我們的祖先天生每隻手就具有六支手指，則現在很可能是使用十二進制（基底 12 ）。若他們天生每隻手僅具有 1 支手指，那很可能我們現在使用的二進制（基底 2 ）來計數。要是這樣，那我們今天學起電腦來，就會容易多了！爲了對電腦能做充分的利用，現在我們必須做許多十進制與二進制間的不便轉換。

不論使用那一種基底，數目系統的結構都是一樣的。例如，十進數（基底 10 ）243.72所代表的意義爲

$$2(10)^2 + 4(10)^1 + 3(10)^0 + 7(10)^{-1} + 2(10)^{-2}$$

同樣地，任一個數目 $a_1a_2a_3 \cdot a_4a_5$ ，若基底爲 b ，則其所代表的數目爲

$$a_1(b)^2 + a_2(b)^1 + a_3(b)^0 + a_4(b)^{-1} + a_5(b)^{-2}$$

例如，若基底爲 7 ，則 423.51 即表

$$4(7)^2 + 2(7)^1 + 3(7)^0 + 5(7)^{-1} + 1(7)^{-2}$$

若基底爲 5 ，則 13.423 表

$$1(5)^1 + 3(5)^0 + 4(5)^{-1} + 2(5)^{-2} + 3(5)^{-3}$$（8.904（基底 10 ））

若欲與 Apple Ⅱ 適當地溝通，那我們至少必須學會三種數目系統：十進制，二進制，與十六進制。十進制以十爲基底，使用 0 ，1 ，2 ，3 ，4 ，5 ，6 ，7 ，8 ，9 等十個符號計算，每逢十就進位，這我想大家都知道。二進制是眞正電腦所使用的數目系統。由于構成電腦之電路均具有雙穩態的特性。因此，目前的數位計算機幾乎都

使用二進制。十六進制是人類對電腦所使用之二進制的一種方便縮寫。其爲電腦與人類之間一種極便捷的轉換方式。

二進制 （Binary）

二進制以二爲基底，僅使用 0 與 1 兩個符號計算，每逢二就進位。因此，二進制的數目寫起來就是一串的 0 和 1 。例如，二進制 1011011 卽代表：

$$\begin{aligned}&1(2)^6 + 0(2)^5 + 1(2)^4 + 1(2)^3 + 0(2)^2 + 1(2)^1 \\&+ 1(2)^0 \\&= 64 + 0(32) + 16 + 8 + 0(4) + 2 + 1 \\&= 91\end{aligned}$$

由此可見，二進數 1011011 卽等於十進數 91 。

同理，二進數 110111 代表十進數 55。因爲，

$$\begin{aligned}&1(2)^5 + 1(2)^4 + 0(2)^3 + 1(2)^2 + 1(2)^1 + 1(2)^0 \\&= 55\end{aligned}$$

以上兩個例子所示卽爲二進數轉換成十進數的過程。反之，若已知十進數欲轉換成其對等二進數，則我們卽必須由可能的最大乘冪開始，每次減去一個 2 的乘冪。例如，欲將十進數 237寫成二進制時，我們首先注意到，$128 = 2^7$ 是 237 所能減去的最大 2 乘冪，故我們將 237 減去 128 ，得到 109 。由於 109 所得減去的最大 2 乘冪爲 $2^6 = 64$ ，故我們再將 $109 - 64 = 45$ 。至此 $237 = 128 + 64 + 45$ 。由於 $45 = 32 + 13$ 且 $13 = 8 + 4 + 1$ 。故我們得

$$\begin{aligned}237 &= 128 + 64 + 32 + 8 + 4 + 1 \\&= 1(2)^7 + 1(2)^6 + 1(2)^5 + 0(2)^4 + 1(2)^3 + 1(2)^2 \\&\quad + 0(2)^1 + 1(2)^0 \\&= 11101101\text{（基底爲 2 ）}\end{aligned}$$

習 題

1. 將下列十進數轉換成二進數：
   (a) 145　　(b) 252
   (c) 201　　(d) 75
2. 將下列二進數轉換成十進數：
   (a) 11011100　　(b) 10110111
   (c) 10010001　　(d) 11001101

Apple Ⅱ 爲數位計算機，因此，其算術均以二進制完成。此外，Apple Ⅱ 亦爲八位元的計算機，這是說，其每一次能同時處理與傳輸八位元的資料。基於此，故 Apple電腦的記憶體亦組織成一系列八位元的方塊。每八位元稱爲一個位元組。每四位元稱爲半個位元組。因此二進數 110111110（＝十進數 190 ）爲一個位元組數目。其低次的半位元組爲 1110 ，高次的半位元組爲 1011 。

或許讀者已算出八位元二進數所能表示的最大數值了！此值爲 11111111 等於十進數 255 。雖然如此，Apple 電腦所能處理的最大數目並不受限於此，在 Apple計算機內，更大的數目可以數個位元組或其它方式（如浮點型式）來表示。

十六進制 （Hexadecimal）

十六進制以 16 爲基底 ，採用 0 ，1 ，2 ，3 ，4 ，5 ，6 ，7 ，8 ，9 ，A ，B ，C ，D ，E ，F 等 16 個符號計數，每逢就進位。因此，在十六進制內，數目 235 表示

$$2(16)^2 + 3(16)^1 + 5(16)^0 = 565$$

爲了避免與十進數混淆在一起，我們特別在十六進數前面加一錢號“$”字首作區別。例如，十六進數 235 以後我們就寫成 $ 235 而 $235 = 565 表示“十六進數 235 等於十進數 565” 。

記住，在十六進制內，十進數 10 ，11 ，12 ，13 ，14 ，15 分別以符號 A ，B ，C ，D ，E ，F 表示。因此，十六進制看起來就像：$ 1 ，$ 2 ，$ 3 ，$ 4 ，$ 5 ，$ 6 ，$ 7 ，$ 8 ，$ 9 ，

$A，$B，$C，$D，$E，$F，$10，$11，$12，$13，$14，$15，$16，$17，$18，$19，$1A，$1B，$1C，$1D，$1E，$1F，$20，$21 ……

> 注意：$24讀成“十六進數二四”而勿讀成“十六進數二拾四”。因爲“二拾四”代表十進制。

十六進數換成十進數的方法與二進數換成十進數的方法類同。唯一不一樣的是，基底是 16，而非 2。舉例，十六進數$2A3E換成十進數的方法爲：

$2A3E

$= 2(16)^3 + 10(16)^2 + 3(16)^1 + 14(16)^0$

$= 2(4096) + 10(256) + 3(16) + 14(1)$

$= 8192 + 2560 + 48 + 14$

$= 10814$

同理

$E38F

$= 14(16)^3 + 3(16)^2 + 8(16)^1 + 15(16)^0$

$= 57344 + 768 + 128 + 15$

$= 58255$

爲了節省時間，我們可以直接查表 A1.1。透過此表格，我們可以輕易將下列十六進數換成其對等十進數：

$1111 等於4096 + 256 + 16 + 1 = 4369

$2202 等於8192 + 512 + 0 + 2 = 8706

$253B 等於8192 + 1280 + 48 + 11 = 9531

$EB29 等於57344 + 2816 + 32 + 9 = 60201

反之，十進數欲轉換成其對等十六進數時，我們首先必須找出十進數所能減去的最大乘冪。例如，欲將十進數 29143轉換成十六進數

表A1-1

	16^3	16^2	16^1	16^0	
$1	4096	256	16	1	$1
$2	8192	512	32	2	$2
$3	12288	768	48	3	$3
$4	16384	1024	64	4	$4
$5	20480	1280	80	5	$5
$6	24576	1536	96	6	$6
$7	28672	1792	112	7	$7
$8	32768	2048	128	8	$8
$9	36864	2304	144	9	$9
$A	40960	2560	160	10	$A
$B	45056	2816	176	11	$B
$C	49152	3072	192	12	$C
$D	53248	3328	208	13	$D
$E	57344	3584	224	14	$E
$F	61440	3840	240	15	$F

時，我們首先發現，這個數最大能減去 $7(16)^3 = 28672$。因此，我們將之減去這個數目：$29143 - 28672 = 471$。接著，再找出 471 最大能減去 $1(16)^2 = 256$。故又求得 $471 - 256 = 215$。繼續，215 最大又能減去 $13(16) = 208$。是以，我們又求得 $215 - 208 = 7$。因此，我們得

$$\begin{aligned} 29143 &= 28672 + 471 \\ &= 7(16)^3 + 256 + 215 \\ &= 7(16)^3 + 1(16)^2 + 208 + 7 \\ &= 7(16)^3 + 1(16)^2 + 13(16)^1 + 7 \\ &= \$71D7 \end{aligned}$$

同理

$$\begin{aligned} 43618 &= 40960 + 2658 \\ &= 10(4096) + 2560 + 98 \\ &= 10(16)^3 + 10(256) + 96 + 2 \\ &= 10(16)^3 + 10(16)^2 + 6(16)^1 + 2(16)^0 \\ &= \$AA62 \end{aligned}$$

若您發覺十進制與十六進制之間轉換太麻煩，那您可使用下面兩

個程式。這兩個程式第一個將一已知十六進數轉換成其對等十進數，第二個則反之將一已知十進數轉換成其對等十六進數。可能的話，您可將兩個程式合併成一個程式。此外，您可自行設計有關一些二進數的轉換程式。

程式A1·1：十六進制轉換十進制

```
1   REM PROGRAM A1.1 (BASE CONVERTER)
2   REM HEX TO DECIMAL
10   INPUT "NUMBER (HEX) ";A$
20 L = LEN (A$):N = 0
30   FOR I = 0 TO L - 1:
     T = ASC ( MID$ (A$,L - I,1))
40   IF T = 36 THEN GOTO 80
50 T = T - 48: IF T > 9 THEN T = T - 7
60 N = N + T * 16 ^ I
70   NEXT I
80   PRINT N
```

程式A1·2：十進制轉換十六進制

```
1   REM PROGRAM A1.2 (BASE CONVERTER)
2   REM DECIMAL TO HEX
10 H$ = "0123456789ABCDEF"
20 BASE = 16:I = 1
30   INPUT "NUMBER (DECIMAL) ";N:N = INT (N)
40 Q = INT (N / BASE)
50 A(I) = N - BASE * Q:I = I + 1
60   IF Q > BASE THEN N = Q: GOTO 40
70 A(I) = Q
80   PRINT "$";
90   FOR J = I TO 1 STEP - 1
100   PRINT MID$ (H$,A(J) + 1,1);
110   NEXT J
```

十六進制與二進制的關係 （Relationship Of Hex To Binary）

0至15 的十進數，每一個均能以一個四位元的二進數（由0000至1111）表示，或由一位十六進數字（$0至$F）表示。0

至255的十進數則可以一個位元組（八位元）的二進數（由00000000至11111111）表示，或由兩位十六進數字（由$00至$FF）表示。

例如，十進數181表示成二進制即爲10110101，表示成十六進制則爲$B5。注意，二進制表示的低次四位元0101，其十六進制值正好是$5，而且高次四位元1011的十六進制值正好爲$B。換言之，一已知二進數欲轉換成對等十六進數時，我們只要將其每四個位元一組，分別代換成其對等十六進數字即可。反之，一已知十六進數欲轉換成其對等二進數時，我們只要分別將每一位十六進數字，分別代換成其對等四位元二進數即可。下面即是一些例子：

例 1：試將 $6C 寫成二進形式

解：因爲 $6 = 0110（二進制）且

$C = 1100（二進制）

故 $6C = 01101100（二進制）

例 2：試將二進數 10111101 寫成十六進制

解：高次四位元 1011 的十六進數爲 $B

低次四位元 1101 的十六進數爲 $D

因此 10111101（二進制）= $BD

習　題

1. 試將下列二進數轉換成十六進數。

(A) 10111001　　(C) 10001111

(B) 10111100　　(D) 11011011

2. 試將下列十六進數轉換成二進數。

(A) $3F　　(C) $2D

(B) $CA　　(D) $E6

通常十六進數與十進數之間的轉換是不須讀者自行去換算的。可由表A1.2直接由十六進位的數字轉換成十進位的數字，表A1.2的

範圍是十六進數的0至FFF，相等於十進數的0至4095。至於超過數字的轉換時，讀者可以利用表A1.3的範圍來補充表A1.2的不足，其間技巧的運算，盼讀者自行實驗。

表A1-2 十六進位－十進位整數換算表

十六進位—十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

十六進位－十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
10	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12	0288	0289	0290	0291	0292		0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13	0304	0305	0306	0307	0308	0	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
20	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

十六進位－十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
60	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1562	1663
68	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

十六進位－十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
70	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72	1824	1825	1826	1827	1828	[illegible]	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73	1840	1841	1842	1843	1844	[illegible]	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
80	2048	2049	2050	2051	2052	[illegible]	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B	2480	2481	2482	2483	2484	24[illegible]	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

十六進位—十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C0	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

十六進位－十進位整數換算表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D0	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3	3376	3377	3378	3379	3380	33	382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4	3392	3393	3394	3395	3396	339	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E0	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE	3808	3809	3810	3811	3812	381	814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC	4032	4033	4034	4035	4036	4037	38	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

表 A1-3　表 A1-2 的範圍擴充表

十六進位—十進位整數換算表

十六進位	十進位	十六進位	十進位
01 000	4 096	20 000	131 072
02 000	8 192	30 000	196 608
03 000	12 288	40 000	262 144
04 000	16 384	50 000	327 680
05 000	20 480	60 000	393 216
06 000	24 576	70 000	458 752
07 000	28 672	80 000	524 288
08 000	32 768	90 000	589 824
09 000	36 864	A0 000	655 360
0A 000	40 960	B0 000	720 896
0B 000	45 056	C0 000	786 432
0C 000	49 152	D0 000	851 968
0D 000	53 248	E0 000	917 504
0E 000	57 344	F0 000	983 040
0F 000	61 440	100 000	1 048 576
10 000	65 536	200 000	2 097 152
11 000	69 632	300 000	3 145 728
12 000	73 728	400 000	4 194 304
13 000	77 824	500 000	5 242 880
14 000	81 920	600 000	6 291 456
15 000	86 016	700 000	7 340 032
16 000	90 112	800 000	8 388 608
17 000	94 208	900 000	9 437 184
18 000	98 304	A00 000	10 485 760
19 000	102 400	B00 000	11 534 336
1A 000	106 496	C00 000	12 582 912
1B 000	110 592	D00 000	13 631 488
1C 000	114 688	E00 000	14 680 064
1D 000	118 784	F00 000	15 728 640
1E 000	122 880	1 000 000	16 777 216
1F 000	126 976	2 000 000	33 554 432

附錄2　造型表

Applesoft 的HPLOT與HPLOT TO 命令提供了一種產生圖像的方式。除此外，還有第二種方法可用，這種方法尤其在顯示常用的圖形是非常有用的。把常用的圖形（ shapes ）組成一個圖形表格，此處我們暫叫“造型表”（ shape table ），有了此表，我們即可用DRAW或XDRAW命令將這些圖形加至一圖像上。使用造型表的理由有下：

⑴造型表一旦準備好後，程式就可一個命令顯示出一個圖形，而不論此圖形有多複雜。

⑵圖形可輕易呈現，然後除去，而不會影響背景圖像。這種能力在圖像動態上極具價值。

⑶圖形也可旋轉與定比例大小。

當然，圖形的使用也有些不便的地方，例如，經重新定比例或旋轉的圖形會有點失眞。不過，最困難的還是在造型表的設計與建立。若用手去做，則工作不僅費勁，且易出錯，所幸，計算機可以替我們完成。不過，條件是必須有適當的程式。目前，此類程式在計算機雜誌上曾經登過，而且市面有售。附錄內程式 2.2 即是其中一例。

緊接下面幾頁即討論組成一個造型表的程序，並爲程式 A2.2 提供一個基礎。不過，使用程式A2.2並不須事先了解以下的這些討論。如果讀者的目的僅在使用這個程式，請您直接參閱“造型表的用法”一節。

圖形與造型表 （Shapes And Shape Tables）

圖形即爲一系列代表線段的寫碼數目，線段定義一個圖形的形式。其可被看成是一群指明構成物體的點與線段欲畫在那裏的指令。

爲了能被存取，以便畫圖，一個圖形可爲一造型表的一部份。造型表的第一部份包含一個定址結構。表格的第一個位元組指出表格內所含的圖形數N，這個N值最大可爲255。表格的第2個位元組未用。緊接的2N個位元組指出由表格的第一個位元組至表格內每一圖形之每一個位元組間的距離。每一個距離（或位移）均以兩個位元組表示，低次位元組存在前。

每一圖形由一系列數碼組成，這些數碼代表圖形的軌跡向量。每一向量即向上、下、左、右移動一個單位距離的命令。此外，向量有兩種型態：畫然後移動的向量，以及不畫只移動的向量。定義圖形時使用8個向量及數碼如表A2.1所示：

表A2-1

不畫而移動向量		畫且移動向量	
（向量）	（數碼）	（向量）	（數碼）
上 o	0	上 •	4
右 o	1	右 •	5
下 o	2	下 •	6
左 o	3	左 •	7

例如：我們定一個起始點，畫出下列向量：6，6，2，6，6，1，1，6，3，6，3，7，0，4，3，5所得圖形如圖A2.1所示。

雖然已知系列寫碼向量，我們即可畫出一個圖形，但造型表呈現出向量的格式對畫圖形命令而言亦非常重要。看起來，將向量數碼依序儲於記憶器內，每一個向量儲存在一個位元組上，似乎很合理，不過實際做起來却很冒險。此外，這樣使用記憶體亦不夠效率，因每一個向量碼數目最多僅用了每一個記憶位元組中的三個位元。

事實上，畫圖形指令每次最多可由造型表的一個位元組讀取三個

向量。不過，並不永遠這麼高效率。無法一次讀取三個向量主要有兩個狀況：

1. 每一向量碼均長三個位元組。若三個向量的最高次位元均不爲零（三個向量數目均大於或等於 4），則這時候三個向量就必須以九個位元加以儲存。一個位元組（八個位元）存不下。

2. 畫圖命令會忽略出現在表格之任意位元組最前面的零，並將全爲零的位元組解釋成一個代表圖形終了的數碼。爲此，零向量（不畫，僅向上移）必須小心地儲存。

我們以定義圖 A2.1 圖形的向量，來舉例說明圖形之向量表格的構成。

在圖 A2.1 的圖形中，開始系列 6，6，2 的二進形式爲 110，110，010。將此三個由右至左依序列出，我們獲得 010，110，110 除去逗號後，得到一八位元數目 10110110。同理，下一個系列 6，6，1 可寫成 01，110，110 或 01110110。再緊接的系列 1，6，3 寫成 11110001。如此，圖形的最初九個向量即可寫碼成 10110110，01110110，11110001 三個位元組。

再緊接三個向量 6，3，7 寫成二進制 110，011，111。由右至左依序列出，數碼變成 111011110。顯然，此超出八位元。因此，我們可將前兩個向量存入位元組內，變成 00011110（前導零忽略），並將第三個向量延至下一個位元組。

再次應考慮的三個向量是 7，0，4。同樣的，這三個向量亦是一個位元組存不下。此外，由於畫圖形的命令忽略前導零。故前兩個向量不能存於一個位元組。我們必須把 7 存成一位元組，而把另外的兩個向量延至下一個位元組。

0，4，3 三個可合併在一個位元組，存成 11100000。最後一個向量 5 則自己存一個位元組。最後，再一個全爲零的位元組代表圖形終了。

如此，圖 A2.1 所示的圖形，其向量碼 6，6，2，6，6，1

，1，6，3，6，3，7，0，4，3，5，即可如下存成八個位元組

二進制　　十進制

10110110 = 182
2　6　6
01110110 = 118
1　6　6
11110001 = 241
3　6　1
00011110 =　30
　3　6
00000111 =　 7
　　7
11100000 = 224
3　4　0
00000101 =　 5
　　5
00000000 =　 0
圖形終了數碼

因而，我們就可構成一個僅含此圖形的簡短造型表：

第1位元組：　1　（圖形個數）
第2位元組：　0　（未用）
第3位元組：　4　（至1號圖形之第一位元組的位移）
第4位元組：　0
第5位元組：182
第6位元組：118
第7位元組：241
第8位元組：　30　（1號圖形碼）

第 9位元組：　7
第 10 位元組：224
第 11 位元組：　5
第 12 位元組：　0　（圖形碼終了）

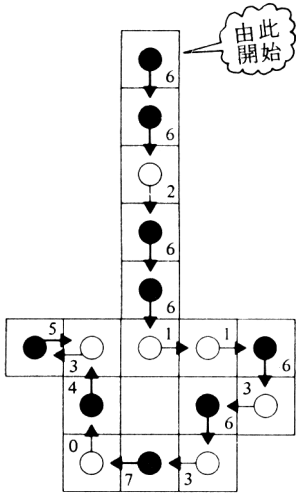


圖 A2-1

程式A2、2 的研究　（Notes On The Program A2.2）

此程式有一部份是與使用者作彼此溝通用的。若程式屬於這一部份的這裏暫先不討論。

程式 A2.2 能建立一個新的圖形表格，或自磁碟將一現有的表格取入記憶體內（行號 40 － 90），不論那一種情況，造型表格均正好位於第 2 高解像繪圖頁的上方。若指明新表格，則程式會騰出記憶空間（行號 230），並存起由表格開始至第一個圖形開始之間的位移（行號 200 － 220）。在與使用者的溝通允許圖形建立方格被畫出後，

程式轉向組合圖形的工作。

一個圖形的設計包含兩個步驟：(1)指出構成圖形的向量，與(2)組合向量碼，獲得造形表的位元組。這兩個步驟在設計圖A2.1的圖形時您都看過了，其就是程式A2.2的基礎。這個程式的最重要部份（去除一些使用者作界面溝通的命令後），以下

```
410   GET V$: PRINT
420   IF, V$ = "Q" THEN POKE ADDR,P:
      POKE ADDR + 1,255: GOTO 520
440   IF V$ = "P" THEN P = 4: GOTO 410
450   IF V$ = "U" THEN V = P
460   IF V$ = "R" THEN V = P + 1
470   IF V$ = "D" THEN V = P + 2
480   IF V$ = "L" THEN V = P + 3
500   POKE ADDR,V: ADDR = ADDR + 1
510 P = 0: GOTO 410
520 ADDR = PEEK (TA + SN * 2 + 2) + 256 *
    PEEK (TA + SN * 2 + 3) + TA:BYTE = ADDR
530 C1 = PEEK (BYTE): IF C1 = 255
    THEN POKE ADDR,0: ADDR = ADDR + 1: GOTO 660
540 C2 = PEEK (BYTE + 1): IF C2 = 255
    THEN POKE ADDR,C1: POKE ADDR + 1,0:
    ADDR = ADDR + 2: GOTO 660
550 C3 = PEEK (BYTE + 2): IF C3 = 255
    THEN POKE ADDR,C1 + 8 * C2: POKE ADDR + 1,0:
    ADDR = ADDR + 2: GOTO 660
560 CODE = C1 + 8 * C2 + 64 * C3
570   IF CODE = 0 THEN POKE ADDR,64:
      POKE ADDR + 1,24: ADDR = ADDR + 1:
      BYTE = BYTE + 2: GOTO 650
580   IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
590   IF CODE < 64 THEN POKE ADDR,CODE:
      BYTE = BYTE + 1: GOTO 650
600   IF CODE < 256 THEN POKE ADDR,CODE:BYTE =
      BYTE + 2: GOTO 650
610 CODE = CODE - 64 * C3
620   IF CODE = 0 THEN POKE ADDR,64:
      POKE ADDR + 1,3:ADDR = ADDR + 1:
      BYTE = BYTE + 1: GOTO 650
630   IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
640   POKE ADDR,CODE:BYTE = BYTE + 1
650 ADDR = ADDR + 1:BYTE = BYTE + 1: GOTO 530
```

當向量由鍵盤打入後，行號500即將向量碼存入一個暫時性的表格內。當您打“Q”代表圖型結束時，程式即將最後向量以及一暫時性的圖形終了數碼（255）存起。

行號 520 將 ADDR 的值令成暫時表格的起點，並將 byte 的值令成圖形的起始位址（在此二者都一樣）。

行號 530 至 550自暫時表格讀取三個連續的向量，若這三個向量中有任一者爲 255（圖形終了電碼），圖形卽結束。否則，行號 560 卽求出三個向量的數碼 CODE 。

若 CODE 的值爲 0（行號 570），則表三個向量的每一個向量均爲不畫，僅上移向量。由於全部爲 0 的位元組會被解釋成圖形終了電碼，因此，我們將連續三個均爲 0 的向量系列，特別擴增成 0 ，0 ，1 ，0 ，3 。如此，原來的上移向量等於分解成了右移(1)與左移(3)兩個向量的合成。0 ，0 ，1 ，0 ，3 五個向量剛好分成兩個位元組：64（0 ，0 ，1 ）與 24（0 ，3 ）

若 CODE 的值介於 1 與 7 間（行號 580），則其必須代表一個非零向量碼緊接兩個零向量碼。由於繪圖指令忽略前導零，故前導零我們留至下個位元組用。

若 CODE 的值介於 8 與 63 間（行號 590 ），則其必須代表一系列第二個不爲零，但第三個爲零的向量。同樣地，這個第三個向量必須留至下個位元組用。

若 CODE 的值介於 64 與 255 間，則自臨時表格所讀的三個向量卽可合成一個位元組（行號 600 ）。

若 CODE 的值大於或等於 256 ，則表示 CODE 就必須要有 9 個位元。這時候，第三個向量必須延至下一個位元組（行號 610 ），且前兩個向量必須再作零向量分析（看是否爲零）（行號 620 － 630）。

在圖形組合完成後，程式卽顯示出結果（行號 670 ），並詢問使用者是否願意將圖形個數 SN 的值進 1 ，並將至下一個圖形的位移存起（行號 680 ）。

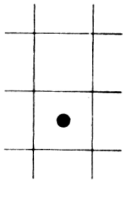
造型表的使用法 （Using Shapes）

造型表中的圖形在能顯示之前，造型表必須事先設計好，並由磁碟存入計算機內，存在一個不會再用作另一種用途的安全記憶區內。由於位址$300（ 768）記憶位置起的記憶空間非常有限，僅能用以儲存一個很小的造型表。因此，較大的造型表必須儲存於記憶體的上半部份（在DOS之下），並將HIMEM 的值設定在表格起始位址之下，以達到保護作用。另一個通常也是安全的記憶區域是在第2頁高解像繪圖頁上。程式A2.2的圖形即存在此處。

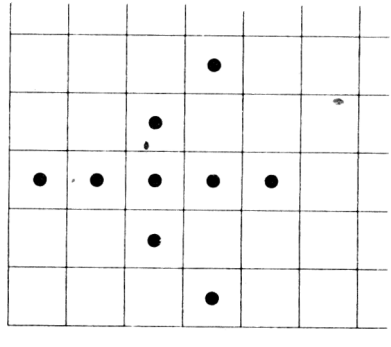
造型表在記憶體內的位置雖然可以任意設定。但其起始位置則必須使繪圖頁命令知道。不論使用DRAW或XDRAW 命令，造型表的位址均存於位址232（存低半部）與233（存高半部 ）兩個記憶位置內，而使這兩個位置含適當的地址是程式設計者的責任。

一般而言，若造型表存在位址A的記憶位置，則位址232的記憶位置將有A－256*(INT(A／256))之內容，且地址233之記憶位置將有INT(A／256)的內容。當表格由磁碟中取入時，這些值即可設定。由此可見，若能在適當的時刻重新設定位址232與233兩個記憶位置的內容，則一個程式將可以同時使用好幾個造型表。

在使用DRAW或XDRAW指令之前，ROT，SCALE，HCOLOR



(a)



(b)

圖A2-2

的值一定要事先設定。您會發現，圖形在旋轉或定比例後會有一點失眞。此外，由於絕大部份圖形都含有垂直線段，所以將發覺，色彩並不容易使用。因此所有顏色中，只有白色爲一般人比較喜歡用。

緊接的簡短程式說明了如何以一個造型表模擬雷達網。這個造型表指定在行號 20 － 30 內。其總共包含兩個圖形：一個畫然後上移向量，以及一群構成一架飛機的向量。這兩個圖形分別如圖 A2.2。當 1 號圖形比例被設定時（ SCALE = 75），其看起來就宛如一條直線，並且被用作雷達網的掃描線。當掃描線掃過雷達網時，兩架飛機的位置卽稍爲改變一下。此程式可進一步擴充成模擬空戰或飛機場之作業情形的遊戲程式。

程式A2·1：雷達

```
1   REM PROGRAM A2.1 (RADAR)
2   REM ILLUSTRATES THE USE OF SHAPES
10   FOR I = 768 TO 782: READ X: POKE I,X: NEXT I
20   POKE 233,3: POKE 232,0
30   HGR2 : ROT= 0: SCALE= 2
40 DT = .1:C = COS (DT):S = SIN (DT):N = 6.4 / DT
50 CX = 140:CY = 96:SC = 1.16:X = 50:Y = 0:FL = 0
60   GOSUB 190
70 X = 30:Y = 0:FL = 0: GOSUB 190
80 A = 175:B = 61:C = 191:D = 113
90   XDRAW 2 AT A,B: XDRAW 2 AT C,D
100   SCALE= 75: FOR I = 0 TO 63
110   ROT= I: XDRAW 1 AT 140,96
120   FOR J = 1 TO 15: NEXT J
130   IF I = 8 THEN SCALE= 2: ROT= 1:
      XDRAW 2 AT A,B:A = A - 3:B = B + 3:
      XDRAW 2 AT A,B: SCALE= 75
140   IF I = 19 THEN SCALE= 2: ROT= 1:
      XDRAW 2 AT C,D:C = C - 3:D = D - 1:
      XDRAW 2 AT C,D: SCALE= 75
150   IF A < 140 THEN TEXT : HOME : END
160   ROT= I: XDRAW 1 AT 140,96
170   NEXT I: GOTO 100
180   DATA 2,0,6,0,8,0,4,0,109,229,240,54,14,4,0
190   FOR J = 1 TO N
200 T = X * C - Y * S:Y = Y * C + X * S:X = T
210 SX = SC * X + CX:SY = CY - Y
220   IF FL = 1 THEN 240
230   HPLOT SX,SY:FL = 1
```

```
240  HPLOT  TO SX,SY
250  NEXT J: RETURN
```

程式A2、2的圖案(圖形組合) (Program A2.2 (Shape Construction))

程式將數個圖形組合成一個造型表，並將整個表以二進檔案的形式儲存在磁碟上。程式當然會提供一個讓您將造型表儲存在磁碟上，然後再叫出的機會。因此，此造型表是分了好幾個段落所組成的。

程式使用以一個顯示在顯示幕上之方格中之點的組合型態定義一個圖形。方格的大小可由使用者指定(最大爲 15 × 15)，當在設計同一大小的圖形時(如英文字母)，能控制方格的大小是非常重要的。

何處欲爲圖形起點的方格同樣是由使用者來決定。圖形的這一點畫在顯示幕上的位置則由繪圖命令指定。若設計的圖形很多時，通常我們都會以同一位置作爲圖形的起點。

欲令指標向上，下，左，右各移動位置時，我們可分別使用U，D，L，R鍵。按P鍵將在目前的指標位置上畫出一點。圖形畫完或您想再試一次時，請按Q鍵。按下Q鍵時，計算機會停一會兒，組合讀者已畫出的圖形，然後再將結果顯示在方格的右邊。

> 注意：若您喜歡以I，J，K，M鍵移動浮標(像在編輯文字幕一樣)，可在行號 450 至 480 換成這些按鍵。

在完成每一圖形後，您可選擇將之併入造型表內，或不要它。然後，您可以繼續再設計另一圖形，改變方格的大小，改變起始點，將現有的造型表存入磁碟，或離開程式，完全決定於讀者的喜好了！

程式A2·2：圖形的組合

```
1   REM PROGRAM A2.2 (SHAPE CONSTRUCTION)
2   REM ASSISTS IN CONSTRUCTING SHAPE TABLES
10   HCOLOR= 3: SCALE= 1: ROT= 0:TA = 24576
20    POKE 768,1: POKE 769,0: POKE 770,4: POKE
      771,0
30    POKE 772,58: POKE 773,36: POKE 774,45: POKE
      775,54: POKE 776,7: POKE 778,0
40    TEXT : HOME : PRINT "CHOOSE:": PRINT
50    PRINT "L --> LOAD A SHAPE TABLE FROM DISK":
      PRINT
60    PRINT "N --> START A NEW SHAPE TABLE":
      PRINT
70    PRINT "YOUR CHOICE "; CHR$ (95);: HTAB 12:
      INPUT A$
80     IF A$ = "N" THEN 180
90    IF A$ < > "L" THEN 40
100    PRINT : INPUT "NAME OF THE TABLE ":N$
110    PRINT CHR$ (4);"BLOAD ";N$;",A$6000"
120 N = ( PEEK (TA + 2) + 256 *
    PEEK (TA + 3) - 2) / 2
130 SN = PEEK (TA): PRINT : IF SN = N THEN PRINT
    "TABLE FULL": END
140    PRINT "THIS TABLE CAN HOLD ";N;" SHAPES":
       PRINT
150    PRINT "IT PRESENTLY HAS ";SN;" SHAPE(S) IN
       IT": PRINT
160    FOR I = 1 TO 2500: NEXT I
170    GOTO 240
180    TEXT : HOME : INPUT "NUMBER OF SHAPES IN
       THIS TABLE: ";N
190    POKE TA,0: POKE TA + 1,0
200 DA= 2 * N + 2
210    POKE TA + 2,DA - 256 * INT (DA / 256)
220    POKE TA + 3,INT (DA / 256)
230    FOR I = TA + 4 TO TA + 2 * N + 3: POKE I,0:
       NEXT I
240    PRINT : PRINT "CHOOSE SIZE OF SHAPE DESIGN
       GRID"
250    INPUT "NUMBER OF COLUMNS (1 - 15) ";C:C =
       10 * C
260    INPUT "NUMBER OF ROWS (1 - 15) ";R:R = 150
       - 10 * R
270    HGR
280    FOR I = R TO 150 STEP 10: HPLOT 0,I TO C,I:
       NEXT I
290    FOR I = 0 TO C STEP 10: HPLOT I,R TO I,150:
       NEXT I
300    IF A = 1 THEN 330
310    HOME : VTAB 21: PRINT "STARTING POSITION?
       LOWER LEFT IS (1,1)"
```

(continued)

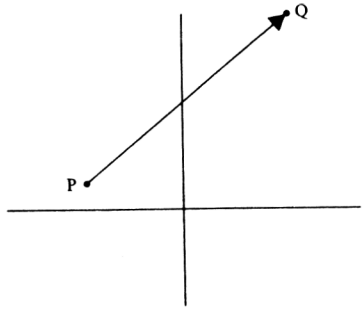
```
320  INPUT "COLUMN ";X1: INPUT "ROW ";Y1
330 X = 10 * X1 - 5:Y = 155 - 10 * Y1
340 ADDR = PEEK (TA + SN * 2 + 2) + 256 * PEEK
    (TA + SN * 2 + 3) + TA
350  POKE 232,0: POKE 233,3
360  XDRAW 1 AT X,Y:P = 0:V = - 1
370  HOME: VTAB 21: PRINT "TABLE CAPACITY: ";N;"
    SHAPES-THIS IS # ";SN + 1
380  VTAB 22: PRINT TAB( 10);"DIRECTION
    (U/D/L/R) "
390  VTAB 23: PRINT "CHOOSE: PLOT (P)"
400  VTAB 24: PRINT TAB( 10);"QUIT THIS SHAPE
    (Q)";
410  VTAB 23: HTAB 35: GET V$: PRINT
420  IF V$ = "Q" THEN POKE ADDR,P: POKE ADDR +
    1,255: GOTO 520
430  XDRAW 1 AT X,Y
440  IF V$ = "P" THEN P = 4:
    FOR I = X - 3 TO X + 3: HPLOT I,Y - 3 TO
    I,Y + 3: NEXT I: XDRAW 1 AT X,Y: GOTO 410
450  IF V$ = "U" THEN V = P:Y = Y - 10: IF Y < R
    THEN Y = Y + 10: PRINT CHR$ (7);:V = - 1
460  IF V$ = "R" THEN V = P + 1:X = X + 10: IF X
    > C THEN X = X - 10: PRINT CHR$ (7);:V = -
    1
470  IF V$ = "D" THEN V = P + 2:Y = Y + 10: IF Y
    > 150 THEN Y = Y - 10: PRINT CHR$ (7);:V =
    - 1
480  IF V$ = "L" THEN V = P + 3:X = X - 10: IF X
    < 0 THEN X = X + 10: PRINT CHR $ (7);:V = -
    1
490  IF V = - 1 THEN XDRAW 1 AT X,Y: GOTO 410
500  POKE ADDR,V:ADDR = ADDR + 1
510  XDRAW 1 AT X,Y:P = 0:V = - 1: GOTO 410
520 ADDR = PEEK (TA + SN * 2 + 2) + 256 * PEEK
    (TA + SN * 2 + 3) + TA:BYTE = ADDR
530 C1 = PEEK (BYTE): IF C1 = 255 THEN POKE
    ADDR,0:ADDR = ADDR + 1: GOTO 660
540 C2 = PEEK (BYTE + 1): IF C2 = 255 THEN POKE
    ADDR,C1: POKE ADDR + 1,0 :ADDR = ADDR + 2:
    GOTO 660
550 C3 = PEEK (BYTE + 2): IF C3 = 255 THEN POKE
    ADDR,C1 + 8 * C2: POKE ADDR + 1,0:ADDR =
    ADDR + 2: GOTO 660
560 CODE = C1 + 8 * C2 + 64 * C3
570  IF CODE = 0 THEN POKE ADDR,64: POKE ADDR +
    1,24:ADDR = ADDR + 1:BYTE = BYTE + 2: GOTO
    650
580  IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
590  IF CODE < 64 THEN POKE ADDR,CODE:BYTE =
    BYTE + 1: GOTO 650
```

```
600  IF CODE < 256 THEN POKE ADDR,CODE:BYTE =
     BYTE + 2: GOTO 650
610 CODE = CODE - 64 * C3
620  IF CODE = 0 THEN POKE ADDR,64: POKE ADDR +
     1,3:ADDR = ADDR + 1:BYTE = BYTE + 1: GOTO
     650
630  IF CODE < 8 THEN POKE ADDR,CODE: GOTO 650
640  POKE ADDR,CODE:BYTE = BYTE + 1
650 ADDR = ADDR + 1:BYTE = BYTE + 1: GOTO 530
660  POKE TA,SN + 1: POKE 233,96: DRAW SN + 1 AT
     200,100
670  HOME : VTAB 22: PRINT "SAVE THIS AS SHAPE
     NUMBER ";SN + 1;"(Y/N)";: INPUT A$
680  IF A$ = "Y" THEN SN = SN + 1: IF SN < N
     THEN DA = ADDR - TA: POKE TA + 2 * SN +
     2,DA - 256 * INT (DA / 256): POKE TA + 2 *
     SN + 3, INT (DA / 256)
690  IF A$ < > "N" AND A$ < > "Y" THEN 670
700  POKE TA,SN
710  TEXT : HOME : VTAB 17: PRINT "CHOOSE:"
720  PRINT TAB( 5);"(1) DRAW ANOTHER SHAPE"
730  PRINT TAB( 5);"(2) CHANGE DOT MATRIX"
740  PRINT TAB( 5);"(3) CHANGE STARTING POINT"
750  PRINT TAB( 5);"(4) SAVE SHAPE TABLE TO
     DISK"
760  PRINT TAB( 5);"(5) LEAVE THE PROGRAM"
770  INPUT A$:A = VAL (A$)
780  IF A < 1 OR A > 5 THEN 710
790  IF SN = N AND A < 4 THEN PRINT "TABLE
     FULL": FOR I = 1 TO 1000: NEXT I: GOTO 710
800  ON A GOTO 270,240,270,820
810  TEXT : HOME : END
820  INPUT "NAME ";N$
830  PRINT CHR$ (4);"BSAVE ";
     N$;",A24576,L";ADDR - 24576
840  GOTO 710
```

附錄3　向　量

一個向量可以看成是一個有方向性的線段。對點P和Q而言，從P到Q的向量可以表示爲$\overrightarrow{PQ}$，或一單獨符號$\vec{v}$。若已知P和Q的座標，如P（a，b）和Q（c，d），則向量$\vec{v}$可以其分量表示。其X軸向量爲$v_x = c - a$，Y軸分量爲$v_y = d - b$。

如圖A3.1所示之P（－3，1）和Q（4，6），向量$\overrightarrow{PQ} = \vec{v}$其分量爲$v_x = 7$和$v_y = 5$。習慣上我們都表示爲$\vec{v}$＝（7，5）。



圖A3-1

雖然（3，7）既代表點P（3，7），亦可代表從O（0，0）到P（3，7）的向量，但我們應該不致搞混。

向量的加算可經由分量而很容易地定義（參閱圖A3.3）。若$\vec{u}$＝（a，b）且$\vec{v}$＝（c，d），則定義$\vec{u} + \vec{v}$＝（a＋c，b＋d）。

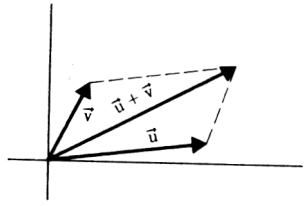


圖 A3-2

同理，向量的減算亦可定義：若$\vec{u}$＝（ a ，b ）且$\vec{v}$＝（ c ，d ），則$\vec{u}-\vec{v}$＝（ a－c ，b－d ）。

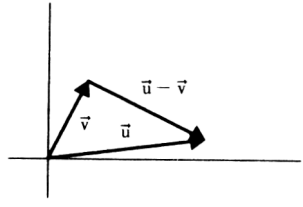


圖 A3-3

向量的純量乘算 （Scalar Multiples Of Vectors）

將一向量乘上一個數目（純量），即爲純量乘算。若$\vec{u}$＝（ a ，b ），則 t$\vec{u}$之積，定義成 t$\vec{u}$＝t（ a ，b ）＝（ ta,tb ）。（ t 爲純量。）若將向量乘上正的純量，結果會把向量本身的長度放大純量那麼多倍，且其方向不變。若將向量乘上負的純量，結果會使向量本身的長度放大純量的絕對值那麼多倍，但方向相反。如圖A3.4 。

直線的向量表示 （Vector Representation Of Lines）

若$\vec{v}$和$\vec{w}$分別爲從O（ 0 ，0 ）到P 和Q點的向量，則$\vec{u}$＝$\vec{w}$－

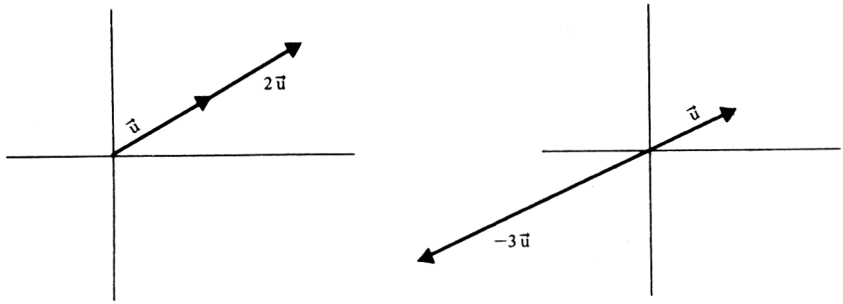


圖 A3-4

$\vec{v}$ 爲從 P 到 Q 點的向量。若 $\vec{x}$ 爲另一個從 O（ 0 ，0 ）到 PQ 直線上某一點的向量，則 $\vec{x} - \vec{v}$ 即爲 $\vec{u}$ 的倍數：$\vec{x} - \vec{v} = t\vec{u}$ ，或 $\vec{x} = \vec{v} + t\vec{u}$（請參閱圖 A3.5 和圖 A3.6 ）。

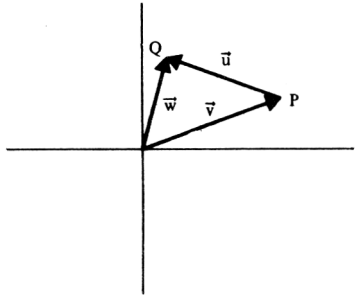


圖 A3-5

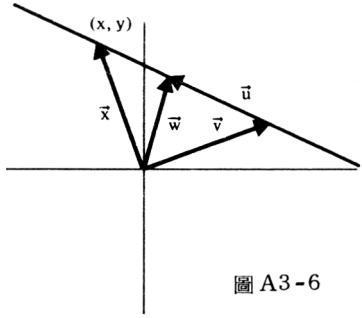


圖 A3-6

例如，設$\vec{v}$＝（10,3），$\vec{w}$＝（1，7）（如圖A3.6），則$\vec{u}=\vec{w}-\vec{v}$＝（1，7）－（10,3）＝（－9，4）。在PQ 線上任選一$\vec{x}$＝（x，y），則$\vec{x}=\vec{v}+t\vec{u}$，或（x，y）＝（10,3）＋t（－9，4）。每選定一t 值，即可得到一位於PQ 線上之一點（x，y）的值。如：

t	x	y
1	1	7
2	－8	11
3	－17	15
0	10	3
－1	19	－1
－2	28	－5

注意，因向量$\vec{u}$有正負兩個方向，故直線亦有正負兩個方向。

三度空間內的向量 （Vectors In Three-Dimensional Space）

前面所提的都是假設向量在二度空間內，我們也可以很容易地的將之擴充至更高度空間。大部分我們都是使用三度空間的向量計算。如向量$\vec{v}$＝（2，1，7）且$\vec{w}$＝（3，2，－5），則

$\vec{v}+\vec{w}$＝（2，1，7）＋（3，2，－5）＝（5，3，2）

$\vec{v}-\vec{w}$＝（2，1，7）－（3，2，－5）

＝（－1，－1，12）

$4\vec{v}$＝4（2，1，7）＝（8，4，28）

通過（2，1，7）和（3，2，－5）兩點的直線可表示成$\vec{x}$＝$\vec{v}+t\vec{u}$，其中$\vec{u}=\vec{w}-\vec{v}$＝（3，2，－5）－（2，1，7）＝（1，1，－12）。因此，此直線其向量型式的方程式爲（x，y，z）＝（2，1，7）＋t（1，1，－12）。

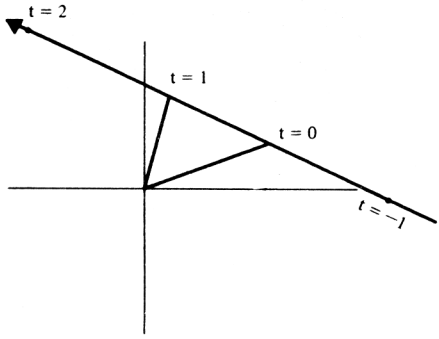


圖 A3-7

向量的點積　（Dot Products）

向量 $\vec{u}$（a，b，c）和 $\vec{v}$ =（d，e，f）的點積，定義爲 $\vec{u}\cdot\vec{v}$ =（a，b，c）·（d，e，f）= ad + be + cf 。例如，若 $\vec{u}$ =（9，1，2），$\vec{v}$ =（3，−5，7），則 $\vec{u}\cdot\vec{v}$ =（9，1，2）·（3，−5，7）= 27 − 5 + 14 = 36 。

在二度空間的向量，其點積之定義亦類似：（a，b）·（c，d）= ac + bd 。例如，（3，5）·（7，−3）= 21 − 15 = 6 。

由這些定義雖可輕易地求出兩向量之點積，但卻無法看出點積的眞正意義。因此，我們另外表示爲 $\vec{u}\cdot\vec{v} = |\vec{u}|\,|\vec{v}|\cos\theta$ 。其中 $|\vec{u}|$ 爲向量 $\vec{u}$ 之大小或長度，$|\vec{v}|$ 爲向量 $\vec{v}$ 的大小或長度，而 θ 爲 $\vec{u}$ 和 $\vec{v}$ 之間的夾角。根據這個解釋，我們可以得到許多有用的應用。在此，我們只需一個：若 $\vec{u}$ 和 $\vec{v}$ 之間的夾角介於 0° 和 90° 之間，則 $\vec{u}$ 和 $\vec{v}$ 的點積爲正。若 $\vec{u}$ 和 $\vec{v}$ 之間的夾角介於 90° 和 180° 之間，則兩者的點積爲負。

θ 的值可以由點積求得：

因爲 $\vec{u}\cdot\vec{v} = |\vec{u}|\,|\vec{v}|\cos\theta$

則 $\cos\theta = \dfrac{\vec{u} \cdot \vec{v}}{|\vec{u}|\,|\vec{v}|}$

且 $\theta = \arccos \dfrac{\vec{u} \cdot \vec{v}}{|\vec{u}|\,|\vec{v}|}$

因 $|\vec{u}|$ 和 $|\vec{v}|$ 恒爲正，故 $\vec{u} \cdot \vec{v} = |\vec{u}|\,|\vec{v}| \cos\theta$ 的符號乃取決於 $\cos\theta$ 的符號。且當 θ 介於 0° 到 90° 之間時，$\cos\theta$ 爲正，當 θ 介於 90° 到 180° 之間時，$\cos\theta$ 爲負。而得證之。

舉例言之，若 $\vec{u} = (1, 3)$，$\vec{v} = (3, -5)$，且 $\vec{w} = (-1, -6)$（如圖 A3.8），則 $\vec{u} \cdot \vec{v} = (1, 3) \cdot (3, -5) = 3 - 15 = -12$ 爲負，乃因 u 和 v 之間夾角介於 90° 和 180° 之間。且 $\vec{u} \cdot \vec{w} = (3, -5) \cdot (-1, -6) = -3 + 30 = 27$ 爲正，因其夾角介於 0° 和 90° 之間。

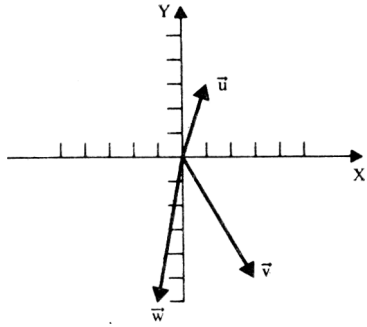


圖 A3-8

向量的交積（Cross Products）

向量交積的定義要比點積麻煩一點。若 $\vec{u} = (a, b, c)$ 且 $\vec{v} = (d, e, f)$，則其交積定義爲 $\vec{u} \times \vec{v} = (bf - ce, cd - af$

，ae − bd ）。注意，$\vec{u} \cdot \vec{v}$為一數字，但$\vec{u} \times \vec{v}$為一向量。此外，$\vec{u} \cdot \vec{v}$在二度、三度空間均有定義，但$\vec{u} \times \vec{v}$僅三度空間的向量有定義。

$$\text{For } \vec{u} = (1, 9, 8) \text{ and } \vec{v} = (3, 0, 7)$$
$$\vec{u} \times \vec{v} = (1, 9, 8) \times (3, 0, 7)$$
$$= (63 - 0, 24 - 7, 0 - 27)$$
$$= (63, 17, -27)$$

交積有何意義呢？我們只對一種應用有興趣：在右手座標系統內，向量$\vec{u}$和$\vec{v}$的交積即爲一垂直於$\vec{u}$和$\vec{v}$構成之平面的向量$\vec{w}$。若您右手食指和中指分別指向$\vec{u}$和$\vec{v}$的方向，則$\vec{w} = \vec{u} \times \vec{v}$的方向即爲大姆指所指的方向。

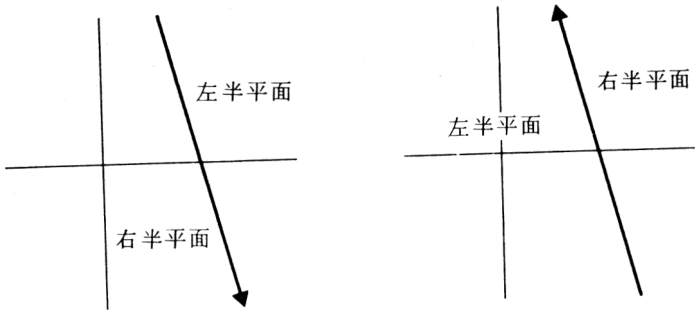
注意，$\vec{u} \times \vec{v}$和$\vec{v} \times \vec{u}$正好反向。例如，若$\vec{u} = (1, 3, 0)$且$\vec{v} = (-3, 7, 0)$，乃位於 xy 平面上，則$\vec{u} \times \vec{v}$即爲（0，0，1）的正倍數。而$\vec{v} \times \vec{u}$則爲（0，0，1）的負倍數。至於在左手系統內（如Apple的顯示幕），若左手中指和食指分別指向$\vec{u}$和$\vec{v}$的方向，則$\vec{u} \times \vec{v}$的方向即爲垂直於$\vec{u} \times \vec{v}$平面的大姆指方向。

半平面 （Half-Planes）

直線（x，y）=（a，b）+ t（c，d）將 xy 平面分成兩個半平面（half-plane）。當面對向量$\vec{v} = (c, d)$的方向時，在左邊的半平面即稱爲左半平面，而在右邊的右半面即稱爲右半平面。特別注意，左、右之分乃視向量$\vec{v} = (c, d)$的方向而定。

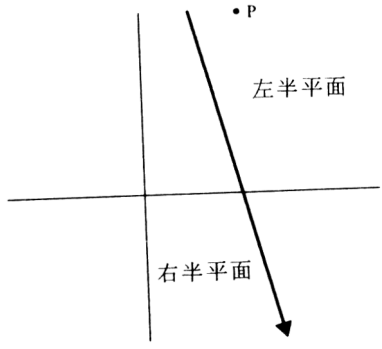
舉例言之，如直線（x，y）=（2，5）+ t（−1，3）分割平面的情形如圖A3.9b 所示。直線（x，y）=（2，5）+ t（1，−3）（同一直線，只是方向相反）分割平面的情形則如圖A3.9a 所示。

半平面主要是在第7章用到，以決定一已知點究竟在那一個半平面內。就直線$\vec{x} = \vec{u} + t\vec{v}$所分割的平面而言，其左，右半平面如圖



圖A3-9

A3.10 所示。欲求出那一半平面含P（ a ， b ）點時，我們先定義 $\vec{v}$ =（ a ， b ）且 $\vec{w}=\vec{r}-\vec{u}$ 。若P點位於左半平面，則 $\vec{v}\times\vec{w}$ 將爲（ 0 ， 0 ， 1 ）的負倍數。



圖A3-10

例如，一直線（ x ， y ）=（ 2 ， 5 ）+ t（ 1 ，－3 ），且P點爲（ 4 ， 8 ），則

$\vec{u}$ =（ 2 ， 5 ， 0 ），$\vec{v}$ =（ 1 ，－3 ， 0 ），$\vec{r}$ =（ 4 ， 8 ， 0 ）

$\vec{w}=\vec{r}-\vec{u}$ =（ 4 ， 8 ， 0 ）－（ 2 ， 5 ， 0 ）

=（ 2 ， 3 ， 0 ）

且

$$\vec{v}\times\vec{w}=(1,-3,0)\times(2,3,0)$$
$$=(0,0,3+6)$$
$$=(0,0,9)=9(0,0,1)$$

其結果爲(0，0，1)的正倍數，即P點位於左半平面，(如圖A3.10)。

注意：以上所言之測試一點位置的方法乃對右手座標系統而言。不過Apple的顯示幕卻是左手座標系統。因此，在Apple顯示幕上分辨點與半平面時，上述的規則必須再修改：$\vec{u}$，$\vec{v}$，$\vec{r}$，$\vec{w}$，與$\vec{P}$的定義同前，但若$\vec{v}\times\vec{w}$結果爲(0，0，1)之負倍數，則P位於左半平面；若$\vec{v}\times\vec{w}$爲(0，0，1)的正倍數，則P位於右半平面。

附錄4　矩陣

所謂矩陣卽是一個矩形的數目陣列（ Array ）。例如，下面例子卽是：

$$\begin{pmatrix} 3 & -9 \\ 4 & 2 \end{pmatrix} 或 \begin{pmatrix} 5 & 8 \\ 2 & 1 \\ 3 & 4 \end{pmatrix}$$

矩陣的用途很廣，而且數學家們目前已發明了許多有關矩陣的公式與技巧。由於我們使用矩陣的目的僅在表示變換，因此，所必須知道有關矩陣與矩陣乘積的知識亦非常有限。

矩陣元素的取名與名稱規則如下之矩陣所示：

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

您可發現，每一矩陣的元素均以兩個註脚（ subscript）代表：第一個註脚表示矩陣元素所在的列數，第二個註脚表示矩陣元素所在位置的行數。例如，a_{24} 表示第 2 列第 4 行上的元素，而 a_{31} 表示第 3 列第 1 行的元素。像 A 這樣具有 3 列與 4 行的矩陣，我們稱爲“3×4”的矩陣。

矩陣的乘積 （ Products Of Matrices ）

矩陣乘積的定義如下：

A　×　B　=　C

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{pmatrix}$$

注意，乘積矩陣 C = AB 之元素 C_{ij} ，正好等於 A 矩陣之第 i 列上的元素與 B 矩陣之第 j 行上的元素，兩兩對應相乘積的和。例如，C_{21} 等於 A 之第 2 列元素（ a_{21} ， a_{22} ）與 B 之第一行元素（ b_{11} ， b_{21} ）兩兩對應相乘積的和。

由於這樣，故矩陣乘積 AB 唯有在 A 之每一列的元素個數與 B 之每一行上的元素個數相同時才有意義。換言之，若 A 爲一 m × n 之矩陣，B 爲一 p × q 之矩陣，則 AB 唯有 n = p 時才可能。AB 兩矩陣相乘後，所得之乘積矩陣的列數正好等於 A 之原有列數，行數等於 B 之原有行數；即 AB 將變成一 m × q 的矩陣，例如：

$$\begin{pmatrix} 1 & 4 & 2 \\ 3 & 0 & -1 \\ 1 & 5 & 2 \end{pmatrix}\begin{pmatrix} 4 & 9 & 8 \\ 3 & -4 & -7 \\ -5 & 0 & 6 \end{pmatrix} = \begin{pmatrix} 6 & -7 & -8 \\ 17 & 27 & 18 \\ 9 & -11 & -15 \end{pmatrix}$$

$$(2,1,3)\begin{pmatrix} 4 & 2 & -5 \\ 8 & 9 & 3 \\ 1 & -4 & 7 \end{pmatrix} = (19,1,14)$$

數學證明的結果，矩陣乘積一般均不適於交換律（即 AB ≠ BA），但却適合於結合律：

$$(A(BC)) = ((AB)C)$$

由於點的座標可認爲是一個 1 × 2 的矩陣。因此

$$(x, y)\begin{pmatrix} 4 & 2 \\ 8 & 9 \end{pmatrix}$$

的乘積顯示矩陣

$$\begin{pmatrix} 4 & 2 \\ 8 & 9 \end{pmatrix}$$

爲每一個（ x ， y ）點與另一點（ 4x + 8y ， 2x + 9y）建立起關係

。例如：

$$(1,-1)\begin{pmatrix}4 & 2\\ 8 & 9\end{pmatrix}=(-4,-7)$$

$$(-2,1)\begin{pmatrix}4 & 2\\ 8 & 9\end{pmatrix}=(0,5)$$

同理，任意矩陣

$$\begin{pmatrix}a & b\\ c & d\end{pmatrix}$$

亦可認爲建立起(x,y)點與$(ax+cy,bx+dy)$之間的關係。這種情況我們稱將(x,y)點變換成$(ax+cy,bx+dy)$點，並稱

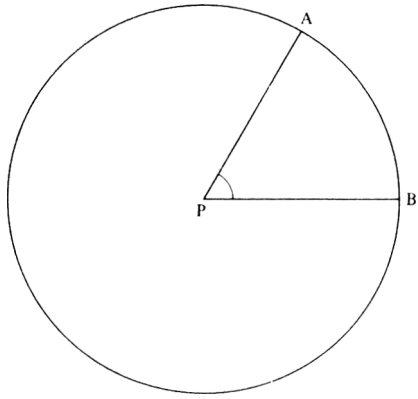
$$\begin{pmatrix}a & b\\ c & d\end{pmatrix}$$

爲變換矩陣。

附錄5　角的弧度測量

在測量一個角時，有很多單位可以使用。一般都是使用度（ degree ）來測量，但這卻不適於科學或工程方面，亦不適用於電腦。此時用弧度（ radian ）來測量將較爲方便。

圖 A5.1 爲一單位圓（半徑爲 1 ），且∠APB 的頂點在圓心，則∠APB 的弧度即 AB 弧的長度。



圖A5-1

因此，弧度的測量亦即測量一單位圓的弧長。

度和弧度之間的關係也就建立在單位圓的圓周長 $C = 2\pi r = 2\pi(1) = 2\pi$ 。因此，一個 360° 的角也就等於 2π 的弧度。亦即 1° 等於 $2\pi / 360$ 。

度轉換成弧度的公式爲：

$$\text{弧度} = \text{度} \times \pi / 180$$

而弧度轉換成度的公式則爲：

$$\text{度} = \text{弧度} \times 180 / \pi$$

表 5.1 爲一些常用角的度－弧度轉換表。

表A5-1

度	弧度
0°	0
30°	$\pi/6 \approx .5236$
45°	$\pi/4 \approx .7854$
60°	$\pi/3 \approx 1.0472$
90°	$\pi/2 \approx 1.5708$
180°	$\pi \approx 3.1416$
270°	$3\pi/2 \approx 4.7124$
360°	$2\pi \approx 6.2832$

在程式裏使用裏弧度時，若能提供一個 π 值將會方便得多。因此，我們可以定義一個變數，如

```
PI = 3.141592654
```

或由 tan(45°) = tan (π / 4) = 1 ，而得

```
PI = 4 * ATN(1)
```

附錄6　封面圖形的程式

底下我們將列出本書封面及封底所有圖形的程式，另外還有圖5.1所顯示圖形的程式。每一個程式都附上參考相關章節的簡短說明。程式的說明都是很簡要的，並假設讀者已熟悉相關章節的觀念和程式。

封面的左上圖（Front Cover:Upper Left）

雖然這個程式很長，不過它的結構很單純。程式長是因爲有很多點需要個別指定。

程式在規劃時，曾對APPLE Ⅱ以其背後左下角爲參考點，而做一次近似測量（以吋爲單位）。有些數據在後來會稍微修改，以改善圖形的外觀。有些測量數據置於DATA陳述語之後，指出欲畫出點的X，Y，Z座標。其他的則需要時再指出。

程式利用行號20－60的投影副程式來得到欲畫出點的顯示幕座標。此副程式於第六章曾介紹過，並用於第六章和第七章的程式。

行號1490－1600控制隱藏線的消除，並利用第七章所提到的“塗黑”法。

變數T有兩種用途。第一，它指出塗黑程式的顯示幕頂Y座標（第一個圖形時T＝0，第二個圖形時T＝40）。第二，在畫出第二個圖形時，T用來跳到程式的其他部分。因此，第二個（即較小的）圖形將比第一個圖形少了若干細部。第三個圖形就少了更多細部，因爲它只是由兩個矩形構成而已（行號1940，1950）。

程式A6·1：封面左上圖；

```
1  REM PROGRAM A6.1  (APPLE II)
2  REM  DRAWS UPPER LEFT IMAGE ON FRONT COVER
10  GOTO 130
20 XE =  - X * S1 + Y * C1
30 YE =  - X * C1 * C2 - Y * S1 * C2 + Z * S2
40 ZE =  - X * S2 * C1 - Y * S2 * S1 - Z * C2
   + RHO
50 SX = D * XE / ZE + CX
60 SY = CY - D * YE / ZE: RETURN
70  GOSUB 20
80  HPLOT SX,SY
90  RETURN
100  GOSUB 20
110  HPLOT  TO SX,SY
120  RETURN
130 RHO = 100:D = 1100:CX = 116:CY = 105
140 THETA = .45:S1 =  SIN (TH):C1 =  COS (TH)
150 PHI = 1.25:S2 =  SIN (PH):C2 =  COS (PH)
160  HGR2 : HCOLOR= 3
170  DATA 17.75,  15.25,  2
180  DATA 17.75,  15.25,  1.25
190  DATA 15,    15.25,  0
200  DATA 0,  15.25,  0
210  DATA 0,  15.25,  4
220  DATA  8.75,  15.25,  4
230  DATA 17.75,  15.25,  2
240  DATA 17.75,  0,  2
250  DATA 17.75,  0,  1.25
260  DATA 17.75,  15.25,  1.25
270  DATA 17.75, 0, 2
280  DATA 8.75, 0, 4
290  DATA 0, 0, 4
300  DATA 0, 15.25, 4
310  READ X,Y,Z: GOSUB 70
320  FOR I = 1 TO 9
330  READ X,Y,Z: GOSUB 100
340  NEXT I
350  READ X,Y,Z: GOSUB 70
360  FOR I = 1 TO 3
370  READ X,Y,Z: GOSUB 100
380  NEXT I
390  IF T <  > 0 THEN 470
400 X = .875
410  FOR I = 1 TO 18 STEP 2
420 Y = 14.625:Z = 4: GOSUB 70
430 Y = 15.25: GOSUB 100
440 Z = 2.5: GOSUB 100
```

(continued)

```
450 X = X + .75
460  NEXT I
470  REM LID
480  DATA 1,1, 1,14.3, 7.75,14.3, 7.75,1, 1,1
490 Z = 4: READ X,Y: GOSUB 70
500  FOR I = 1 TO 4
510  READ X,Y: GOSUB 100
520  NEXT I
530 Y = 0:X = 11:Z = 3.5: GOSUB 70
540 Y = 15.25: GOSUB 100
550 X = 12.25:Z = 3.22: GOSUB 70
560 Y = 0: GOSUB 100
570  REM KEYBOARD
580 X = 13.25:Z = 3.125:Y = 3: GOSUB 70
590 Y = 12.5: GOSUB 100
600  IF T <  > 0 THEN 670
610 X = 13.75: GOSUB 100
620 Y = 3: GOSUB 100
630  FOR Y = 3 TO 11.77 STEP .73
640 X = 13.25: GOSUB 70
650 X = 13.75: GOSUB 100
660  NEXT Y
670 X = 14:Z = 2.88:Y = 2.625: GOSUB 70
680 Y = 12.5: GOSUB 100
690  IF T <  > 0 THEN 760
700 X = 14.5: GOSUB 100
710 Y = 2.625: GOSUB 100
720  FOR Y = 2.625 TO 11.425 STEP .73
730 X = 14: GOSUB 70
740 X = 14.5: GOSUB 100
750  NEXT Y
760 X = 14.75:Z = 2.635:Y = 2.7
770  GOSUB 70
780 Y = 12.2: GOSUB 100
790  IF T <  > 0 THEN 860
800 X = 15.25: GOSUB 100
810 Y = 2.7: GOSUB 100
820  FOR Y = 2.7 TO 12.2 STEP .73
830 X = 14.75: GOSUB 70
840 X = 15.25: GOSUB 100
850  NEXT Y
860 X = 15.5:Z = 2.39:Y = 2.7: GOSUB 70
870 Y = 12.2: GOSUB 100
880  IF T <  > 0 THEN 970
890 X = 16: GOSUB 100
900 Y = 2.7: GOSUB 100
910 Y = 2.7:X = 15.5: GOSUB 70
920 X = 16: GOSUB 100
930  FOR Y = 3.7 TO 11.2 STEP .73
940 X = 15.5: GOSUB 70
```

```
950 X = 16: GOSUB 100
960  NEXT Y
970 X = 16.25:Z = 2.145:Y = 4.7: GOSUB 70
980 Y = 10.45: GOSUB 100
990  IF T <  > 0 THEN 1070
1000 X = 16.75: GOSUB 100
1010 Y = 4.7: GOSUB 100
1020 X = 16.25: GOSUB 100
1030  FOR Y = 2.7 TO 3.45 STEP .1
1040 X = 16.25: GOSUB 70
1050 X = 16.75: GOSUB 100
1060  NEXT Y
1070  DEF  FN Z(X) = 5.94 - .2 * X
1080  IF T <  > 0 THEN 1480
1090  REM A
1100 Y = 7:X = 9.95:Z =  FN Z(X): GOSUB 70
1110 X = 9.75:Z =  FN Z(X): GOSUB 100
1120 Y = 6.7: GOSUB 100
1130 X = 9.95:Z =  FN Z(X): GOSUB 100
1140  REM PP
1150  FOR LY = 7.2 TO 7.7 STEP .5
1160 X = 10.05:Y = LY:Z =  FN Z(X): GOSUB 70
1170 X = 9.75:Z =  FN Z(X): GOSUB 100
1180 Y = LY + .3: GOSUB 100
1190 X = 9.95:Z =  FN Z(X): GOSUB 100
1200 Y = LY: GOSUB 100
1210  NEXT LY
1220  REM L
1230 Y = 8.2: GOSUB 70
1240 X = 9.65:Z =  FN Z(X): GOSUB 100
1250  REM  E
1260 Y = 8.8:X = 9.95:Z =  FN Z(X): GOSUB 70
1270 Y = 8.45: GOSUB 100
1280 X = 9.75:Z =  FN Z(X): GOSUB 100
1290 Y = 8.8: GOSUB 100
1300 X = 9.95:Y = 8.45:Z =  FN Z(X): GOSUB 100
1310  REM   ]
1320 Y = 9: GOSUB 70
1330 X = 9.65:Z =  FN Z(X): GOSUB 100
1340 Y = 9.25: GOSUB 70
1350 X = 9.95:Z =  FN Z(X): GOSUB 100
1360  REM BOX
1370 Y = 5.5:X = 10.4:Z =  FN Z(X): GOSUB 70
1380 Y = 9.5: GOSUB 100
1390 X = 9.4:Z =  FN Z(X): GOSUB 100
1400 Y = 5.5: GOSUB 100
1410 X = 10.4:Z =  FN Z(X): GOSUB 100
1420  REM  APPLE
1430  HCOLOR= 5
1440  FOR X = 9.8 TO 10.1 STEP .1:Z =  FN Z(X)
```

```
1450 Y = 6: GOSUB 70
1460 Y = 6.3: GOSUB 100
1470  NEXT X
1480  REM  BLACKOUT
1490 X = 8.75:Y = 3.4:Z = 4: GOSUB 20:P1 = SX:P2
     = SY
1500 Y = 12.4: GOSUB 20:Q1 = SX:Q2 = SY
1510 X = 0: GOSUB 20:R1 = SX:R2 = SY
1520 SL = (Q2 - P2) / (Q1 - P1)
1530  HCOLOR= 4
1540  FOR X = P1 TO Q1
1550 Y = SL * (X - P1) + P2
1560  HPLOT X,T TO X,Y
1570  NEXT X
1580 SL = (R2 - Q2) / (R1 - Q1)
1590  FOR X = Q1 TO R1:Y = SL * (X - Q1) + Q2
1600  HPLOT X,T TO X,Y: NEXT X
1610  REM MONITOR
1620  HCOLOR= 3
1630 X = 8.75:Y = 3.4:Z = 4: GOSUB 70
1640 Y = 12.4: GOSUB 100
1650 Z = 10: GOSUB 100
1660 Y = 3.4: GOSUB 100
1670 Z = 4: GOSUB 100
1680  IF T <  > 0 THEN 1780
1690  REM DIALS
1700 X = 8.75:Y = 11:Z = 6.5: GOSUB 20
1710  FOR TH = 0 TO 6.3 STEP .4
1720  HPLOT SX,SY TO SX + 4 *  COS (TH),SY + 4
      *  SIN (TH)
1730  NEXT TH
1740 Z = 8: GOSUB 20
1750  FOR TH = 0 TO 6.3 STEP .4
1760  HPLOT SX,SY TO SX + 4 *  COS (TH),SY + 4
      *  SIN (TH)
1770  NEXT TH
1780 Y = 3.9:Z = 4.5: GOSUB 20:P1 = SX:P2 = SY
1790 Y = 9.9: GOSUB 20:Q1 = SX:Q2 = SY
1800 Z = 9.5: GOSUB 20:R1 = SX:R2 = SY
1810 Y = 3.9: GOSUB 20:T1 = SX:T2 = SY
1820  HPLOT P1,P2 TO Q1,Q2 TO R1,R2 TO T1,T2 TO
      P1,P2
1830 Z = 4:Y = 12.4: GOSUB 70
1840 X = 0: GOSUB 100
1850 Z = 10: GOSUB 100
1860 Y = 3.4: GOSUB 100
1870 X = 8.75: GOSUB 100
1880 Y = 12.4: GOSUB 70
1890 X = 0: GOSUB 100
1900  IF T <  > 0 THEN 1940
1910  REM  RETURN FOR SECOND IMAGE
```

(continued)

```
1920 T = 40: RESTORE :RHO = 435:CX = 140:CY =
     64: GOTO 310
1930  REM  FAKE THIRD IMAGE
1940  HPLOT 143,55 TO 149,57 TO 149,59 TO 143,58
      TO 143,55
1950  HPLOT 144,56 TO 144,52 TO 148,53
      TO 148,57
]
```

封面的右上圖（Front Cover:Upper Right）

程式A6.2和程式7.6幾乎完全相同，所不同的是使用的函數（行號50）以及產生表面上之點的FOR-NEXT迴路的極限值（行號90，110）。由於畫出的表面所跨越的X和Y值非常寬，再加上有關的投影及隱藏的計算，因此程式畫完整個圖形大約需要40分鐘。

程式A6·2：封面右上圖；餘弦函數的表面

```
1   REM PROGRAM A6.2  (COSINE SURFACE)
2   REM DRAWS UPPER RIGHT IMAGE ON FRONT COVER
10   HGR2 : HCOLOR= 5:CX = 140:CY = 96
20 RHO = 80:D = 750
30 THETA = .4:S1 =  SIN (TH):C1 =  COS (TH)
40 PHI = 1.25:S2 =  SIN (PH):C2 =  COS (PH)
50   DEF  FN Z(X) =  COS (.06 * (X * X + Y * Y))
60   DIM YN(280),YX(280)
70   FOR I = 0 TO 279:YN(I) = 191: NEXT I
80   REM  POINT GENERATING LOOP
90   FOR X = 22 TO  -55 STEP  -1
100 FL = 0
110   FOR Y =  - 40 TO 19
120 Z =  FN Z(X)
130   GOSUB 190
140   NEXT Y,X
150   HCOLOR= 2
160   HPLOT 0,0 TO 278,0 TO 278,191 TO 0,191 TO
      0,0
170   END
180   REM  PLOTTING SUBROUTINE
190 XE =  - X * S1 + Y * C1
200 YE =  - X * C1 * C2 - Y * S1 * C2 + Z * S2
210 ZE =  - X * S2 * C1 - Y * S2 * S1 - Z * C2 +
    RHO
220 SX = D * (XE / ZE) + CX
230 SY =  - D * (YE / ZE) + CY
```

```
240  IF FL = 0 THEN FL = 1:F = 0: GOTO 360
250 DX = OX - SX: IF DX = 0 THEN DX = 1
260 SL = (OY - SY) / DX:YP = OY
270  FOR XP =  INT (OX) + 1 TO SX
280 FG = 1
290 YP = YP + SL
300  IF XP < 0 OR XP > 279 THEN FG = 0:F = 0:
     GOTO 350
310  IF YP > 191 OR YP < 0 THEN FG = 0:F = 0
320  IF YP <  = YN(XP) THEN 370
330  IF YP >  = YX(XP) THEN 420
340 F = 0
350  NEXT XP
360 OX = SX:OY = SY: RETURN
370 YN(XP) = YP
380  IF FG = 0 THEN 410
390  IF F = 0 THEN  HPLOT XP,YP:F = 1
400  HPLOT  TO XP,YP
410  IF YP < YX(XP) THEN 350
420 YX(XP) = YP
430  IF FG = 0 THEN 460
440  IF F = 0 THEN HPLOT XP,YP:F = 1
450  HPLOT TO XP, YP
460  GOTO 350
]
```

封面的左下圖（Front Cover;Lower Left）

這個圖形主要是用來輔助說明多變數微積分（如偏導函數，方向導函數）的一些觀念。不過除了教學用途之外，圖形本身也是很有趣的。

本程式大部分與程式A6.2類似。雖然定義表面的函數（行號 70－110 ）是要複雜多了，不過點和隱藏線的計算則幾乎都相同。程式首先畫出較近的部分，接著畫出垂直平面，最後再畫出較遠的部分。平面另外使用一個投影副程式，且其隱藏線陣列的更新也和表面的控制方式不同（行號 340 ）。原因乃在於當正畫出平面之時，陣列YX(I)不必更新。

此程式畫出整個圖形須要大約25 分鐘的時間，因此請您保持耐性，稍安勿躁。

程式A6·3：封面左下圖；表面／平面

```
1  REM PROGRAM A6.3  (SURFACE/PLANE)
2  REM DRAWS LOWER LEFT IMAGE ON FRONT COVER
10  HGR2 : HCOLOR= 5:CX = 140:CY = 96
20 RHO = 80:D = 750
30 THETA = .4:S1 =  SIN (TH):C1 =  COS (TH)
40 PHI = 1.25:S2 =  SIN (PH):C2 =  COS (PH)
50  DIM YN(280),YX(280)
60  FOR I = 0 TO 279:YN(I) = 191: NEXT I
70  DEF  FN W(X) = 3 *  EXP ( - .1 * ((Y + 10)
     ^2) - .02 * ((X + 10) ^2))
80  DEF  FN R(X) = 4 *  EXP ( - .1 * ((X + 10)
     ^2) - .3 * ((Y - 6) ^2))
90  DEF  FN S(X) = 3 *  EXP ( - .1 * ((Y + 8)
    2) - .02 * ((X - 10) ^2))
100 DEF  FN U(X) = - 6 *  EXP ( - .09 * (X - 5)
    ^2 - .09 * (Y - 7) ^2)
110 DEF  FN Z(X) =  FN U(X) +  FN R(X) +  FN
    W(X) +  FN S(X)
120 XS = 8:YS =  - 8:X2 =  - 9:Y2 = 8
130 SP = (Y2 - YS) / (X2 - XS)
140  HCOLOR= 5
150  REM CLOSER PART OF SURFACE
160  FOR X = 14 TO  - 18 STEP  - 1:FL = 0
170 YL = YS + (X - XS) * SP
180  FOR Y = YL TO 17 STEP .5
190 Z =  FN Z(X): GOSUB 490
200  NEXT Y,X
210  REM CURVE OF INTERSECTION
220 FL = 0
230  FOR X = 14 TO  - 18 STEP  - .4
240 Y = YS + (X - XS) * SP:Z =  FN Z(X): GOSUB
    490
250  NEXT X
260  REM PLANE
270  HCOLOR= 3
280  FOR X = 14 TO  - 18 STEP  - .4
290 Y = YS + (X - XS) * SP:Z = 5
300 FL = 0: GOSUB 780
310  IF SX < 0 OR SX > 279 THEN 390
320  IF SY < 0 THEN SY = 0
330  HPLOT SX,SY TO SX,YN(SX) - 2
340  FOR I = PX TO SX:YN(I) = SY: NEXT I
350 Z =  - 7:FL = 1: GOSUB 780
360  IF SY > 191 THEN SY = 191
370  IF SY > YX(SX) + 2 THEN  HPLOT SX,SY TO
     SX,YX(SX) + 2
380 PX = SX
```

(continued)

```
390  NEXT X
400  REM FARTHER PART OF SURFACE
410  HCOLOR= 5
420  FOR X =  - 5 TO  - 18 STEP  - 1
430 FL = 0
440  FOR Y =  - 13 TO 0 STEP .5
450 Z =  FN Z(X): GOSUB 490
460  NEXT Y,X
470  END
480  REM PROJECTION AND HIDDEN LINES
490 XE =  - X * S1 + Y * C1
500 YE =  - X * C1 * C2 - Y * S1 * C2 + Z * S2
510 ZE =  - X * S2 * C1 - Y * S2 * S1 - Z * C2
    + RHO
520 SX = D * (XE / ZE) + CX
530 SY =  - D * (YE / ZE) + CY
540  IF FL = 0 THEN FL = 1:F = 0: GOTO 660
550 DX = OX - SX: IF DX = 0 THEN DX = 1
560 SL = (OY - SY) / DX:YP = OY
570  FOR XP =  INT (OX) + 1 TO SX
580 FG = 1
590 YP = YP + SL
600  IF XP < 0 OR XP > 279 THEN FG = 0:F = 0:
     GOTO 650
610  IF YP > 191 OR YP < 0 THEN FG = 0:F = 0
620  IF YP <  = YN(XP) THEN 670
630  IF YP >  = YX(XP) THEN 720
640 F = 0
650  NEXT XP
660 OX = SX:OY = SY: RETURN
670 YN(XP) = YP
680  IF FG = 0 THEN 710
690  IF F = 0 THEN  HPLOT XP,YP:F = 1
700  HPLOT  TO XP,YP
710  IF YP < YX(XP) THEN 650
720 YX(XP) = YP
730  IF FG = 0 THEN 760
740  IF F = 0 THEN  HPLOT XP,YP:F = 1
750  HPLOT  TO XP,YP
760  GOTO 650
770  REM PROJECTION - FOR PLANE
780 XE =  - X * S1 + Y * C1
790 YE =  - X * C1 * C2 - Y * S1 * S2 + Z * S2
800 ZE =  - X * S2 * C1 - Y * S2 * S1 - Z * C2
    + RHO
810 SY =  - D * (YE / ZE) + CY
820  IF FL = 1 THEN  RETURN
830 SX = D * (XE / ZE) + CX
840  RETURN
]
]
```

封面的右下圖（Front Cover;Lower Right）

此圖形將一個由極座標方程式R = cos(2θ) 所定義之區域，與一個邊界圓組合而成。極座標方程式在第五章曾簡要的提過，而邊界圓上的點也是利用第五章所提到的反複（ recursive ）法計算出來的。

程式A6·4：封面右下圖；魔咒標誌

```
1  REM PROGRAM A6.4  (POLAR HEX SIGN)
2  REM DRAWS LOWER RIGHT IMAGE ON FRONT COVER
10  HGR2 :SC = 1.16
20 CX = 140:CY = 96
30 C =  COS (.1):S =  SIN (.1)
40  REM POLAR "ROSE"
50  FOR TH = 0 TO 6.3 STEP .05
60 R = 88 *  COS (2 * TH)
70  HCOLOR= 7: IF R < 0 THEN  HCOLOR= 5
80 X = CX + SC * R *  COS (TH)
90 Y = CY + R *  SIN (TH)
100  HPLOT 140,96 TO X,Y
110  NEXT TH
120  REM BOUNDARY CIRCLES
130  HCOLOR= 6
140  FOR R = 90 TO 95 STEP .8:X = R:Y = 0
150  FOR I = 0 TO 63
160 T = X * C - Y * S
170 Y = X * S + Y * C:X = T
180  IF I = 0 THEN  HPLOT CX + SC *
     X,CY - Y
190  HPLOT  TO CX + SC * X,CY - Y
200  NEXT I,R
```

封底的左圖（Back Cover;Left）

此程式乃利用造型表字型，以及將造型表文字印在高解像度圖形顯示幕上的副程式（在第三章曾討論過。）行號60之DATA 陳述語和這些資料元素的總數TL（行號100 ），是用來指出整個圓輪其五個扇形區之每一扇形面積。例如第一個扇形面積即爲整個圓輪面積的

20 / TL = 20 / 135 ≈ 0.14 倍。

行號 70 的 DATA 陳述語乃指出在畫每一扇形區時所使用的顏色。

行號 120 ，170 － 190，則使第三個扇形區特別突出。若行號 120 另外定義爲 N = 2 ，則最突出的將爲第二個扇形區。餘則類推。

行號 60 － 110 亦可加以修改，使各扇形區的大小和顏色能由使用者隨時指定，即程式具有對談（interactive）的特性。如此除了標名的設定之外，程式都能正常動作。因爲標名長度的變化使程式很難適當控制其位置，以致程式無法完全對談化。不過您還是可以試試看。

程式A6·5：封底左圖；圓形圖表

```
1   REM PROGRAM A6.5  (PIE CHART)
2   REM DRAWS LEFT IMAGE ON BACK COVER
10   PRINT  CHR$ (4): "BLOAD SHAPE ALPHABET,
     A$6000"
20   POKE 232,0: POKE 233,96
30   SCALE= 1: ROT= 0:SC = 1.16
40 X = 0:Y =  - 75:PI = 3.14159266
50 C =  COS (.01):S =  SIN (.01)
60   DATA 20,50,30,10,25
70   DATA 3,6,5,2,1
80   FOR I = 1 TO 5
90   READ A(I)
100 TL = TL + A(I)
110   NEXT I
120   HGR2 :N = 3
130   FOR I = 1 TO 5
140 FI = (TL * FI + A(I)) / TL
150   READ K: HCOLOR= K
160 CX = 140:CY = 96
170   IF I <  > N THEN 200
180 CX = CX + 11 *  SIN ((ST + FI) * PI)
190 CY = CY + 11 *  COS ((ST + FI) * PI)
200   FOR TH = ST * 628 TO FI * 628
210 T = X * C - Y * S:Y = X * S + Y * C:X = T
220   HPLOT CX,CY TO CX + SC * X,CY - Y
230   NEXT TH
240 ST = FI
250   NEXT I
260 HT = 25:VT = 22
```

```
270 ST$ = "MISCELLANEOUS": GOSUB 370
280 HT = 34:VT = 9
290 ST$ = "HOUSING": GOSUB 370
300 HT = 1:VT = 2
310 ST$ = "TRANSPORTATION": GOSUB 370
320 HT = 4:VT = 14
330 ST$ = "FOOD": GOSUB 370
340 VT = 21:HT = 5
350 ST$ = "CLOTHING": GOSUB 370
360  END
370 HT = 7 * (HT - 1):VT = 8 * VT - 1
380  FOR I = 1 TO  LEN (ST$)
390 CH =  ASC ( MID$ (ST$,I,1)) - 32
400  IF CH = 0 THEN 420
410  XDRAW CH AT HT,VT
420 HT = HT + 7
430  NEXT I
440  RETURN
```

封底的右圖（Back Cover;Right）

這個程式乃是由數位繪圖機所畫出之 7.1圖的基礎。其差別僅在於用繪圖指令（ HGR ，HPLOT ）來取代相對應的繪圖機指令。

此程式很長卻很單純。端點線的座標是由行號 360 開始的DATA陳述語所提供的。資料被讀入後即送到幾個副程式中之一，每個都是利用透視投影副程式（行號 80 － 120 ）。行號 290 開始的副程式畫出一連串N個可見的線段。行號140開始的副程式利用第七章提到的二分找尋法來消除隱藏線。定義窗口的一連串線段則由行號 810開始的副程式畫出。此副程式設定變數X，Y，Z來定義窗口的左下角。同理，920 － 1010 的副程式畫出車庫門，且設定X，Y，Z定義車庫門的右下角。

程式提供的資料和使用的隱藏線技巧（二分找尋法）將限制觀測點的選擇。或許您想嘗試一些其他的RHO，THETA，和PHI的值（行號20，30，40），不過請小心，這些選擇的範圍是有限制的。

程式A6·6：封底右圖；房屋

```
1  REM PROGRAM A6.6 (HOUSE)
2  REM DRAWS RIGHT IMAGE ON BACK COVER
3  REM THIS IS ALSO FIGURE 7.1
10  HGR2 : HCOLOR= 3
20 RHO = 100:D = 390:CX = 100:CY = 150
30 THETA = .7:S1 =  SIN (TH):C1 =  COS (TH)
40 PHI = 1.3:S2 =  SIN (PH):C2 =  COS (PH)
50  GOTO 350
60  REM PERSPECTIVE PROJECTION
70  READ X,Y,Z
80 XE =  - X * S1 + Y * C1
90 YE =  - X * C1 * C2 - Y * S1 * C2 + Z * S2
100 ZE =  - X * S2 * C1 - Y * S2 * S1 - Z * C2
    + RHO
110 SX = D * XE / ZE + CX
120 SY = CY - D * YE / ZE: RETURN
130  REM HIDDEN LINE ROUTINE
140  GOSUB 70:XP = SX:YP = SY
150  GOSUB 70:XQ = SX:YQ = SY
160  GOSUB 70:XV = SX:YV = SY
170  GOSUB 70:XH = SX:YH = SY
180 V1 = (XH - XV) / 2:V2 = (YH - YV) / 2
190 U1 = XP - XQ:U2 = YP - YQ
200 XT = XV + V1:YT = YV + V2
210  FOR I = 2 TO 7
220 VI =  SGN ((U2 * (XV - XQ) - U1 * (YV - YQ))
    * (U2 * (XT - XQ) - U1 * (YT - YQ)))
230 V1 = V1 / 2:V2 = V2 / 2
240 XT = XT + VI * V1:YT = YT + VI * V2
250  NEXT I
260  HPLOT XV,YV TO XT,YT
270  RETURN
280  REM VISIBLE LINE ROUTINE
290  FOR I = 1 TO N
300  GOSUB 70
310  IF I = 1 THEN  HPLOT SX,SY
320  HPLOT  TO SX,SY
330  NEXT I: RETURN
340  REM DATA FOR HOUSE
350 N = 11: GOSUB 290
360  DATA -11,-20,23,  6,-20,9,  6,-20,0
370  DATA 6,32,0,  0,32,0,  0,32,20
380  DATA 0,0,20,  0,0,13,  0,32,13
390  DATA 6,32,9,  6,-20,9
400 N = 2: GOSUB 290
410  DATA 6,32,0,  6,32,9
420 N = 2: GOSUB 290
430  DATA 6,0,0,  6,0,9
440 N = 3: GOSUB 290
450  DATA 0,32,0,  0,0,0,  6,0,0
460 N = 5: GOSUB 290
```

```
470  DATA 0,32,0,  -11.5,32,0,  -11.5,33.5,0
480  DATA -11.5,33.5,33,  -11.5,32,33
490  GOSUB 140: HPLOT XP,YP TO XQ,YQ
500  DATA -11.5,32,33,  -11.5,32,0
510  DATA 0,32,20,  -14,32,31.5
520  GOSUB 160
530  DATA -14,0,31.5,  -14,32,31.5
540  GOSUB 140: HPLOT XP,YP TO XQ,YQ
550  DATA 0,0,20,  -14,0,31.5
560  DATA -11,-20,23,  -11,0,23
570 N = 2: GOSUB 290
580  DATA -11.5,33.5,0,  -16.5,33.5,0
590 N = 2: GOSUB 290
600  DATA -11.5,33.5,33,  -16.5,33.5,33
610  GOSUB 140: HPLOT XP,YP TO XQ,YQ
620  DATA -16.5,33.5,0,  -16.5,33.5,33
630  DATA -28,32,0,  -16.5,32,0
640  GOSUB 160
650  DATA -28,32,20,  -14,32,33
660 N = 2: GOSUB 290
670  DATA -28,32,20,  -28,32,0
680  GOSUB 140
690  DATA 6,-20,9,  6,32,9
700  DATA 0,0,0,  0,0,10
710  FOR I = 1 TO 5: READ X,Y,Z: GOSUB 810:
     NEXT I
720  DATA 0,5,13.5,  0,14,13.5,  0,23,13.5
730  DATA 0,14,2,  0,23,2
740  FOR I = 1 TO 2: READ X,Y,Z: GOSUB 920:
     NEXT I
750  DATA 6,-11,0,  6,-1.5,0
760 N = 4: GOSUB 290
770  DATA 0,6,0,  0,6,6.5
780  DATA 0,9,6.5,  0,9,0
790  END
800  REM WINDOWS SUBROUTINE
810  GOSUB 80: HPLOT SX,SY
820 Y = Y + 4: GOSUB 80: HPLOT  TO SX,SY
830 Z = Z + 5.5: GOSUB 80: HPLOT  TO SX,SY
840 Y = Y - 4: GOSUB 80: HPLOT  TO SX,SY
850 Z = Z - 5.5: GOSUB 80: HPLOT  TO SX,SY
860 Y = Y + 2: GOSUB 80: HPLOT SX,SY
870 Z = Z - 5.5: GOSUB 80: HPLOT  TO SX,SY
880 Y = Y - 2:Z = Z - 2.75: GOSUB 80: HPLOT
    SX,SY
890 Y = Y + 4: GOSUB 80: HPLOT  TO SX,SY
900  RETURN
910  REM GARAGE DOOR SUBROUTINE
920  GOSUB 80: HPLOT SX,SY
930 Z = Z - 7: GOSUB 80: HPLOT  TO SX,SY
940 Y = Y - 7.5: GOSUB 80: HPLOT  TO SX,SY
```

(continued)

```
950 Z = Z - 7: GOSUB 80: HPLOT  TO SX,SY
960 Z = Z + 5:Y = Y + 1: GOSUB 80: HPLOT SX,SY
970 Z = Z + 1: GOSUB 80: HPLOT  TO SX,SY
980 Y = Y + 5.5: GOSUB 80: HPLOT  TO SX,SY
990 Z = Z - 1: GOSUB 80: HPLOT  TO SX,SY
1000 Y = Y - 5.5: GOSUB 80: HPLOT  TO SX,SY
1010  RETURN
```

圖5.1（Figure 5.1）

微電腦無法直接畫出一條曲線，我們只能利用畫一連串的短直線來達成。本程式說明了如何使一連串的曲線（圓）近似於另一曲線（心臟圖）。

程式共個別畫出 19 個圓。每個圓的圓心都是位於一參考（或基準）圓。此外，每個圓都會通過位於基準圓上的共同點P。而這 20 個點（P點和19 個圓心）等距離分佈在基準圓的圓周。不過基準圓並不畫出來。

程式的行號 60 − 120 用來確認 19 個圓心（CX，CY），它是利用反複法計算出來的。而行號 140 − 210畫出每個經過P點的圓，這也是利用反複法計算而得的。

不過圖5.1的程式和本程式稍有不同，因爲它是用數位繪圖機畫出來的，因此本程式的繪圖指令必須換成相對應的繪圖機指令。除此之外，兩個程式大致是相同的。

程式A6·7：圖 5.1；心臟圖

```
1  REM PROGRAM A6.7  (FIGURE 5.1)
2  REM DRAWS A CARDIOID AS THE
3  REM ENVELOPE OF CIRCLES
10 PI = 3.14159266
20 X0 = 25:Y0 = 0:XN = X0:YN = Y0
30 C =  COS (.1):S =  SIN (.1):N = 64
40 C1 =  COS (PI / 10):S1 =  SIN (PI / 10)
50  HGR2 : HCOLOR= 3
60  FOR I = 1 TO 19
70 T = XN * C1 - YN * S1
```

```
80 YN = XN * S1 + YN * C1:XN = T
90 CX = 140 + XN:CY = 96 - YN
100 X = XO - XN:Y = YO - YN
110  GOSUB 140
120  NEXT I
130  END
140  FOR J = 1 TO N
150 T = X * C - Y * S
160 Y = Y * C + X * S:X = T
170 SX = X + CX:SY = CY - Y
180  IF FL = 0 THEN  HPLOT SX,SY
190  HPLOT  TO SX,SY:FL = 1
200  NEXT J
210  RETURN
```

附錄7　APPLE指令一覽表

APPLESOFT 指令：

ABS	HTAB	REM
AND	IF	RESTORE
ASC	IN#	RESUME
AT	INPUT	RETURN
ATN	INT	RIGHT$
CALL	INVERSE	RND
CHR$	LEFT$	ROT=
CLEAR	LEN	RUN
COLOR=	LET	SAVE
CONT	LIST	SCALE=
COS	LOAD	SCRN(
DATA	LOG	SGN
DEF	LOMEM:	SHLOAD
DEL	MID$	SIN
DIM	NEW	SPC(
END	NEXT	SPEED=
EXP	NORMAL	SQR
FLASH	NOT	STEP
FN	NOTRACE	STOP
FOR	ON	STORE

FRE	ONERR	STR$
GET	OR	TAB(
GOSUB	PDL	TAN
GOTO	PEEK	TEXT
GR	PLOT	THEN
HCOLOR=	POKE	TO
HGR	POP	TRACE
HGR2	POS	USR
HIMEM:	PRINT	VAL
HLIN	PR#	VLIN
HOME	READ	VTAB
HPLOT	RECALL	WAIT
		XDRAW

DOS 指令

APPEND	CHAIN	INIT	POSITION	SAVE
BLOAD	CLOSE	LOAD	READ	UNLOCK
BRUN	DELETE	LOCK	RENAME	VERIFY
BSAVE	EXEC	OPEN	RUN	WRITE

中英文名詞對照

A

address	地址 19,20
addressing	定址 22
animation	動態 19,247～250,253

B

binary	二進制 4,5,40,223～234
binary file	二進檔案 61
binary search	二分找尋 192,208,302
bit	位元 19,249
bit pattern graphics	點圖形圖樣 28,29,55～66
blackout	塗黑 189～191,291
byte	位元組 21,249

C

cardioid	心臟圖 305
circles	圓 10,113～119
clipping	截割 78～82
coordinate plane	平面座標 140
coordinate systems	座標系統 139
cos	餘弦函數 93
cross product	交積 280,281
curve	軌跡，曲線 10,133,134

D

dot product 點積 183,279

E

edge array 邊陣列 184,200

ellipse 橢圓 119～126

F

focus 焦點 120

G

GR 低解像度 21,23,27

graphics pages 繪圖頁，圖形頁 45

graphics screen 繪圖顯示幕 80

graphics tablet 繪圖板 239

H

half-plane 半平面 195,281～283

hexadecimal 十六進制 19,20,21,247,249～254

hidden lines 隱藏線 174～219,291,302

high resolution graphics 高解像度繪圖 21,33

hi res graphics 高解像度繪圖 45

homogeneous coordinates 均質座標 97,99,143

hyperbola 雙曲線 131,132

I

identity 本體 86

inverse 逆（矩陣） 147,148,155

K

K 即等於1024 19

keyboard 鍵盤 50

L

left-hand system 左手系統 149,153,155,158,283

light pen 光筆 239

lines 直線 34,79,110～113,276
long programs 冗長程式 51
lo res graphics 低解像度繪圖 38

M

matrix 矩陣 85,285～287
memory map 記憶體位置表 42,43
memory move 記憶體的搬移 41

N

negatives of images 黑白反倒圖像 49
normal 法線，垂直 182
normal array 法線陣列 182,183,200
nybble 半位元組（＝4個位元） 23,249

O

orientation 方位 174

P

paddles 搖桿 237,238
page 2 of text 第二文字頁 47
parabola 拋物線 128～131
parametric equations 參數方程式 126,127
perspective 透視，透視法 151,160,162
phi 希臘字母 ϕ 141
pixel 光點 3,34,50
plotter 繪圖機 241
polar coordinates 極座標 132
printer 列表機，印字機 243
priority 優先順序，優先權 203～205
product 乘積 285,286
projection 投影 150～153

projection plane 投影平面 142,143,150～153
pushbuttons 按鈕 238

R

radian 弧度 94,289,290
raster 光域 3
reading 讀取 41,237
recursive calculation 重複計算 116
rectangular coordinates 直角座標 113,154
reflection 對稱 89,91,109,146
resolution 解像度，解析度 32,33,52
rho 希臘字母 ρ 141
right-hand system 右手系統 139,140,281
rotation 旋轉 91～93,101,145,146,157,158

S

scaling 比例 77
screen 顯示幕，螢幕 77,78,79
separating plane 隔離面 194,203
sequential transformations 順序變換 98,99,147,148
shape 圖形 13,262
shape tabel 造型表 13,67,68,231～233,261～270
shear 切變 91,93
simulation 模擬 5
sin 正弦函數 93
soft switches 軟性開關 37～39
sounds 聲音 15
spherical coordinates 球體座標 140～142

surface 表面 169,214
surface array 表面陣列 180,181,182,198

T

text 文字 21,22,23
theta 希臘字母 θ 92,141
transformations 變換 83,143
translation 位移 84,97,103,109,146,154

V

vectors 向量 275～283
vertex array 頂點陣列 178,180,197
viewpoint 觀測點 142～144,160,161,162,174,191, 302
viewport 觀測台 82,106

復 文 書 局發行圖書目錄

※ 大 專 用 書 ※
經濟、企管、會統

編號	書名	編(譯)者	定價
706	增訂新編國際貿易實務（1979版）	陳高昂	精330.平290.
	增訂新編國際貿易實務（1980版）	〃	精400.平360.
707	作業研究-決策分析	謝茂良	精254.平214
708	包裝管理入門	李春明	66
709	實驗設計	吳宗正	精170.平132
710	線性規劃	陳光隆	110
711	經濟學	李裕寬	精210.平160
712	統計學(一)	吳宗正	220
713	統計學(二)	〃	182

※ 高 商 用 書 ※

710	最新商業應用文	本局編輯部	60.
711	標準書法範本	本局編輯部	20.
712	注音成語大辭典	陳君引	精160.平120
713	高商國際貿易實務(上)	陳高昂	84
714	高商國際貿易實務(下)	〃等	92

※ 衛 生 、 獸 醫 、 其 他 ※

801	一般外科學	石羽飛	106
802	兒童疾病症狀	林澄潭	88
803	獸醫外科手術學	李振湘	92
804	船舶阻力與推進	郭啓祿	100
805	來自風城	黃淑惠	特 52
806	化大地爲春風	黃淑惠	〃 52
807	杜鵑花城的故事	(臺大)	〃 38
808	最新日本口語文法	桑維孝	〃 40
809	演講與致詞	陳君引	〃 40
810	離騷新詮	林振玉	精92.平特60
811	道德學說評述	林顧祥	平特40
812	王陽明論（小字本）	林振玉	平特60.
813	王陽明論（大字本）	〃	80.
	王陽明論（精裝本）	〃	110.
814	夢 判 斷	蘇長虹	42.
815	樂樂唱唱	林適可	48.
816	智慧的啓發與培育	林適可	平特68.
817	體育常識	郭祝府	10.
818	混凝土薄殼	江鴻信	120.
819	汽車修護與故障探索	賴耿陽	112.

※ 各 種 考 試 用 書 ※

901	地理學通論輯要（中小學教師檢定）	李漢宗	40.
902	土地法精粹（地政特考）	鄭啓輝	120.
903	電工電匠考試問題解析	曾亞文	84.
	升二專，三專，師大教育學院試題用書全套		
904	1.機工科試題	黃慕香	58.
905	2.電工科〃	〃	58.
906	3.電子科〃	〃	58.
907	4.汽車修護科〃	〃	58.
908	5.綜合商科〃	〃	52.
909	6.土木建築科〃	〃	52.
910	7.工業工程工教組〃	〃	58.
911	8.機械製圖科〃	〃	52.
912	9.化工科〃	〃	58.
913	10.各商工科共同科目〃	〃	40.
914	熱機學試題精要	詹滄田	60.
915	高工數學精華(大中國)(一)	施添德	32.
916	高工數學精華(大中國)(二)	施添德	32.
917	〃 (三)(四)	〃	各42.
918	高工力學精華	施啓輝	70.
919	高工機構學重點	戴丁元	40.
920	高工熱力學精要	〃	64.
921	高工金屬材料重點整理	施啓輝	40.
922	高工簡明機械工作法	戴丁元	52.
923	高工物理精華	林明達	42
924	高工數學指引 1～6冊	劉滄垣	74.
925	躍昇三民主義綜論	丁大欽	72.
926	躍昇三民主義附衝刺試題	〃	112.
927	躍昇三民主義衝刺試題	〃	40.
928	建築營造法題解	許學貞	96.

※ 青 少 年 科 學 文 庫 ※

951	20萬個爲什麼？ 吳自強編 蘇石山校正	
	〃 ①（天文部份之一）	特 54.
	〃 ②（天文氣象之二）	〃 58.

直接郵購九折優待，請利用免郵費劃撥
帳號 32104 復文書局（向就近郵局辦理）

地址：臺南市東門路421巷28號
電話：（062）370003·386937

復文書局發行圖書目錄

※ 大專用書 ※
機械、材料、力學、冶金

編號	書名	編(譯)者	定價
544	機械工學概論	賴耿陽	精140.平100.
545	超硬工具	〃	精130.平 92.
546	機械構造力學	〃	精168.平128.
547	機械用電子技術①	賴耿陽	96.
548	機械用電子技術②	〃	82.
549	機械用電子技術③	〃	72.
550	機械活用手册	徐景福	62.
551	機械計測學	賴耿陽	82.
552	CNC車床操作實務及程式設計	李哲文	84.
553	金屬加工學	郭正雄	156.
554	材料試驗	賴耿陽	90.
555	自動化機器設計與製作	賴耿陽	精144.平104.

※ 食品及農業用書 ※

編號	書名	編(譯)者	定價
601	食品化學(上)	吳文成	64.
602	食品化學(下)	〃	78.
603	分析化學(上)	蔡瑞瑜	74.
604	分析化學(下)	〃	64.
605	食品化學附實習(上)	吳文成	100.
606	食品化學附實習(下)	〃	102.
607	食品化學實驗法(上)	〃	36.
608	食品化學實驗法(下)	〃	38.
609	食品化學實驗法(全)	〃	74.
610	食品酵素學	陳自珍	精218.平178.
611	應用食品化學上册	林耕年	特 58.
612	應用食品化學下册	〃	78.
613	應用食品化學(全)	〃	136.
614	農用微生物學(全)	〃	76.
615	食品微生物學理論與操作法(新修訂本)	〃	58.
616	食品工業微生物學(全)	邱健人	(增訂本)120.
617	食品微生物(新修訂本)	林耕年	112.
617-1	食品微生物附實習法	〃	全 162.
618	食品加工 第一册(新修本)(合實習)	〃	104.
619	食品加工 第二册(新修本)(合實習)	〃	110.
620	食品加工 第三册(新編合實習)	〃	164.
621	農產加工(全)	〃	64.
622	水產加工(全)	〃	56.
623	畜產加工(全)·(新增訂新編附實習)	〃	108.
624	園產加工(全)	〃	96.
625	釀造學(全)	〃	76.
626	食品加工機械(增訂新編)	林書成	全 178.
626-1	食品加工機械(增訂新編)	林書成	上 120.下 60.
627	食品品質管制(增訂新編)	邱健人	112.
628	食品乾燥(全)	〃	48.
629	食品加工手册	徐進財	精270.平230.
630	新品質管制概論	〃	48.
631	食品營養	林耕年	100.
632	食品添加物(一)磷酸鹽	邱健人	120.
633	食品生物化學	〃	98.
634	食品營養與調製法	林耕年	72.
635	冷凍食品學	徐進財	84.
636	罐頭製造與檢驗法	林金泉	精162.平122.
637	蔬菜與水果商品之製造	陳自珍	精190.平150.
638	實用食品製造法(新增訂)	葉健安	92.
639	實用食品製造講義	葉健安	126.
640	食品化學加工綜典(上)	林耕年	精346.
641	食品營養調配指南	林耕年	126.
642	高農農業概論	蔡瑞瑜(全)	62.
643	土壤肥料(合實習)全	郭周武	110.
644	水土保持	〃	64.
645	醃漬物處方秘典	林耕年	54.
646	食品製造科學	鄭川	72.
647	食品加工實習(一)(二)	郭政文	各 36.
648	食品加工實習(三)(四)	陳永年	各 48.
649	食品加工實習(五)(六)	林耕年	各 36.

※ 大專用書 ※
經濟、企管、會統

編號	書名	編(譯)者	定價
701	廻歸分析理論與應用	吳宗正	精216.平176.
702	數理統計導論	伊榮泰	精190.平150.
703	機率入門與統計上的應用(增訂本附習題解答)	黃銘欽	精228.平188.
704	或然率導論	宋振春	102.
705	最新世界主要及著名商港	林若平	精172.平132.

直接郵購九折優待，請利用免郵費劃撥
帳號32104 復文書局(向就近郵局辦理)

地址：臺南市東門路421巷28號
電話：(062) 370003·386937

APPLE Ⅱ電腦製圖特點：

本書會使我們的電腦進入繪圖的領域。雖然僅介紹 Apple Ⅱ的繪圖設計，但也適合其它型電腦的繪圖。它也爲便利高階語言程式者有效的發揮更提供了我們對電腦繪圖的研究及更熟悉其觀念和技巧。

本書不僅說明了 Apple Ⅱ的繪圖能力外，也介紹了電腦繪圖所應具備最重要的數學基礎和程式設計技巧。而我們只須稍懂BASIC 語言和稍對電腦繪圖有認識即可。本書的特色包羅萬象包括：

——它包含了 80 個以上利用 BASIC語言在電腦上立即執行的程式，以說明繪圖的觀念。

——它從最基本來描述，讀者只需有小學的數學基礎。

——它完整地描敍了 Apple Ⅱ的繪圖特色，並告訴讀者如何用自個的電腦來設計出最好的圖案。

——它提供如何以平面圖形成立體圖、如何安排隱藏線與隱藏面，以及圖樣的比例大小、旋轉，與平移等技巧的探討。

——它也介紹動態圖案變化設計的技術。

究必印翻　有所權版

APPLE II 電腦製圖

王定揚
周健行
編譯

旗文書局